UNDERSTANDING VISUAL ANALYSIS PROCESSES FROM USER INTERACTIONS USING VISUAL ANALYTICS

A Dissertation
Presented to
The Academic Faculty

by

Yi Han

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology
December 2016

UNDERSTANDING VISUAL ANALYSIS PROCESSES FROM USER INTERACTIONS USING VISUAL ANALYTICS

Approved by:

Professor Gregory D. Abowd, Advisor
School of Interactive Computing
Georgia Institute of Technology

Professor John Stasko, Advisor
School of Interactive Computing
Georgia Institute of Technology

Professor Alex Endert
School of Interactive Computing
Georgia Institute of Technology

Professor James D. Foley
School of Interactive Computing
Georgia Institute of Technology

Professor Rahul C. Basole
School of Interactive Computing
Georgia Institute of Technology

Professor David Gotz
School of Information and Library Science
The University of North Carolina at Chapel Hill

Date Approved: November 10th 2016

ACKNOWLEDGEMENTS

I would like to first thank my advisors, Gregory D. Abowd and John Stasko, for their intellectual enlightenment and continuous support. They have both been tremendous role models on how to be a great professor. I am very grateful and honored to have had this opportunity to learn from them. I would also like to thank my thesis committee, Alex Endert, James D. Foley, Rahul C. Basole, and David Gotz, for your feedback, support, and participation in the process. Their comments helped improve my work. Moreover, I would like to thank my former advisor, Rosa Arriaga. She provided me immense support and guidance in the early years of my study.

Next, I would like to thank Johanna Lantz and the team at the Center for Discovery for their generous support over my research studies and the development of the CfD visualization. It was a great pleasure and honor working with them and I really hope the visualization application can continue to help their team excel at data analytics.

I also have to thank the fantastic group of faculty, students, and staff at both the Ubicomp research group and the GT Visualization lab. They have not only brought me entertaining social activities that balanced my work life but also provided stimulating intellectual discussions. Their company has been an important and mostly delightful part of my experience.

Last but not least, I need to thank my parents. They have always been my best supporters and motivators.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

BAs Behavior Specialists and Behavior Analysts, p. 173.

CfD The Center for Discovery, p. 173.

UI User Interface, p. 5.

SUMMARY

Understanding the visual analysis process taken by people when using a visualization application can help its designers improve the application. This goal is typically achieved by observing usage sessions. Unfortunately, many visualization applications are now deployed online so their use is occurring remotely. These remote usages make it very difficult for designers to directly observe usage sessions in person. A solution to the problem is to analyze interaction logs.

While interaction logs are easy to collect remotely and at scale, they can be difficult to analyze because they require an analyst to make many difficult decisions about event organization and pattern discovery. For example, which events are irrelevant to the analysis and should be removed? Which events should be grouped because they are related to the same feature? Which events lead to meaningful patterns that help to understand user behaviors? An analyst needs to be able to make these decisions to identify different types of patterns and insights based on an analysis goal. If the analysis goal changes during the process, these decisions need to be revisited in order to obtain the best analysis results.

Because of the subjective nature of the analysis process and such decisions, flexibility is required so the process cannot be fully automated. Every decision requires additional effort from an analyst that could reduce the practicality of the analysis process. Therefore, an effective interaction analysis method needs to balance the tradeoffs of flexibility and practicality to best support analysts.

Visual analytics provides a promising solution to this problem because it leverages human's broadband visual analysis abilities with the support of computational

methods. For flexibility, the interactive visualizations can ensure an analyst can dynamically adjust decisions in every step of the process to maximize the variety of patterns that could be identified. For practicality, visualizations can help speed up the data inspection and decision-making process while computational methods can reduce the labor in efficiently extracting potentially useful patterns.

Therefore, in this thesis I employ visual analytics in a visual interaction analysis framework to achieve flexibility and practicality in the visual analysis process for identifying patterns in interaction logs. I evaluate the framework by applying it to multiple visualization applications to assess the effectiveness of the analysis process and the usefulness of the patterns discovered.

CHAPTER I

INTRODUCTION

1.1 Background

Designers of visualization applications usually seek to understand the visual analysis process of their users to improve the applications. Which features are the users most interested in? Which visual analysis methods do users frequently take? Are there any usability issues? These questions can be answered by observing how users are operating the visualization applications. However, nowadays many visualization applications are deployed online so usage may be occurring remotely that makes it difficult to directly observe users' interactive activities.

A promising alternative is to log and analyze users' interactions. Interactions can be easily logged remotely and at scale. The challenge is in the analysis of the log data afterwards. For example, irrelevant events may need to be removed from the analysis because they might overshadow otherwise obvious patterns. But which events are irrelevant to the current line of analysis? Sometimes events should be categorized into new abstract representations to better address an analysis goal. For example, both Pohl et al. [34] and Guo et al. [16] used Yi et al.'s [50] interaction categories to organize log events into abstract categories such as Select and Filter. But how should events be categorized for the current line of analysis? Furthermore, sequential patterns of events can reveal visual analysis methods, which are useful for understanding higher-level behaviors. But how should these patterns be defined and identified in the current line of analysis? All these questions require an analyst's subjective assessment of the analysis goal to make the best decisions accordingly.

Because of this subjectivity, the analysis process cannot be fully automated. An analyst need *flexibility* to be able to organize the interaction events in different ways for extracting meaningful patterns based on the analysis goals. This process typically involves many analysis decisions and a laborious manual inspection of the interaction events. Therefore, the flexibility comes with a tradeoff of human labor, which reduces the *practicality* of the process, especially when the tasks are repeated many times for multiple rounds of analyses.

To keep both flexibility and practically in the process, I applied visual analytics to this problem. Human vision is estimated to have a large information bandwidth of nearly 10 Mbps that enables rapid visual analysis [27]. It is effective in detecting patterns through preattentive processing when the information is presented visually in a certain way [45]. Therefore, the use of effective visualizations is the first step in reducing an analyst's effort in manually inspecting interaction data. However, when a large amount of data is presented, patterns can still easily be hidden from an analyst. In these cases, we can apply an automated algorithm to help extract potentially meaningful patterns. The patterns extracted could then be presented visually to an analyst for rapid determination of which ones might be meaningful to the analysis. For example, as related interactions that represent an activity often occur together, a sequential pattern mining algorithm can be applied to discover such patterns. Then, a visual analytics system can visualize these frequent interaction patterns to help an analyst decide which patterns are meaningful and useful to the analysis goal.

Therefore, in this thesis, I develop a visual interaction analysis framework that employs visual analytics to provide flexibility and practicality in the analysis process for identifying patterns in interaction logs. The framework can be used to augment observation studies to improve the understanding of the visual analysis process.

1.2 Thesis Statement

> *Understanding visual analysis processes from user interactions can be practically achieved through a flexible, analyst-driven process using visual analytics.*

1.3 Research Questions

Based on the thesis statement, a number of related research questions then follow:

A. How do we provide flexibility in the analysis process for identifying patterns?

In advance, an analyst should have an idea about which patterns exist in the data. This question is not only about to what degree can analysts find all the patterns they wish to discover, but also whether they can effectively identify patterns that were not initially anticipated. Moreover, a flexible framework should support finding a variety of patterns. Therefore, different analysts should be able identify different sets of patterns that are meaningful to them.

B. Which types of insights can analysts gain from the identified patterns?

The patterns identified need to be useful to an analyst for this framework to be valuable. Patterns can be useful for a variety of reasons. For example, a pattern that shows an incorrect usage of a feature reveals design flaws of a visualization application. The question is which types of insights can be gained from examining the usage patterns?

C. How do we provide analysts the capability to practically identify and analyze patterns?

The analysis process is impractical if it is too laborious. I aim to require a "reasonable" amount of human labor in the framework to identify a set patterns that are meaningful and useful to analysts. By "reasonable" I mean the analysis process does not require an excessive amount of human labor that results in the cost outweighing the benefit of the analysis.

1.4 Research Contribution

I make the following contributions with this work.

1. A visual analytics framework for providing *flexibility* and *practicality* in the analysis process for extracting patterns from user interactions.

2. A demonstration of the *utility* and *generalizability* of the framework through multiple applications and user studies.

1.5 Thesis Organization

After this chapter, this thesis is organized as follows. Chapter 2 is the Related Work where I show the variety of techniques used for identifying patterns from user interactions in the research literature and what they lack in supporting the analysis process. In Chapter 3, I present a set of interaction analysis tasks and define a method for organizing events to extract patterns. This method is implemented with a visual interaction analysis framework for practical implementation. In Chapter 4, I present an implementation of the framework from the perspective of its most important component—a visual interaction analysis system called IntiVisor. In Chapter 5, I evaluate the flexibility, utility, and practicality of the framework. I conclude this work with a discussion in the last chapter.

CHAPTER II

RELATED WORK

2.1 Overview

Research in many domains has analyzed interactions to identify patterns. In this chapter, I will start with the definition of what I mean by interaction events and who I am referring to as an analyst. Next, I will discuss analysis techniques frequently used by researchers for analyzing interaction logs. Last, I will summarize the limitations of the related work and conclude with the need of a visual interaction analysis framework.

2.2 Terminology

2.2.1 Interaction Event

In this work, I narrowly define an interaction event to occur when a person interacts with a user interface (UI) element to issue a command. As shown in Figure 1, an interaction event has many attributes that can be categorized into two types: activity and metadata.

Typically, when analyzing interaction events, the focus is on the interaction activity, which includes three components: manipulation, parameter, and target. The interaction "clicking the zoom in button" can be seen as having the "click" as the manipulation, "zoom level (e.g., 2)" as the parameter, and "zoom in button" as the target. The manipulation is "how" a user interacted with an application. It is often considered unimportant in interaction analysis as it does not directly indicate what the command issued by the user was. The target encodes information about with "what" UI component was interacted. The parameter supplies any additional information about "what" the interaction does. For example, the parameter can be a status of the visualization after the interaction, such as the zoom level that increases every time the zoom in button is clicked. Alternatively, the parameter can be provided separately by a user. For example, if a user typed a search term to conduct a search, the search term will be the user-provided parameter.

Some attributes of an interaction event can be seen as its metadata or context. These attributes can describe the "when," "who," and "where" properties of the event. The time and user of an interaction cover the first two properties. Other metadata of an interaction show "where" this interaction was used. They can be for example, the usage session information, the version of the visualization application or the occasion the application was used. These attributes are typically used to provide context to the interaction activity during the analysis.

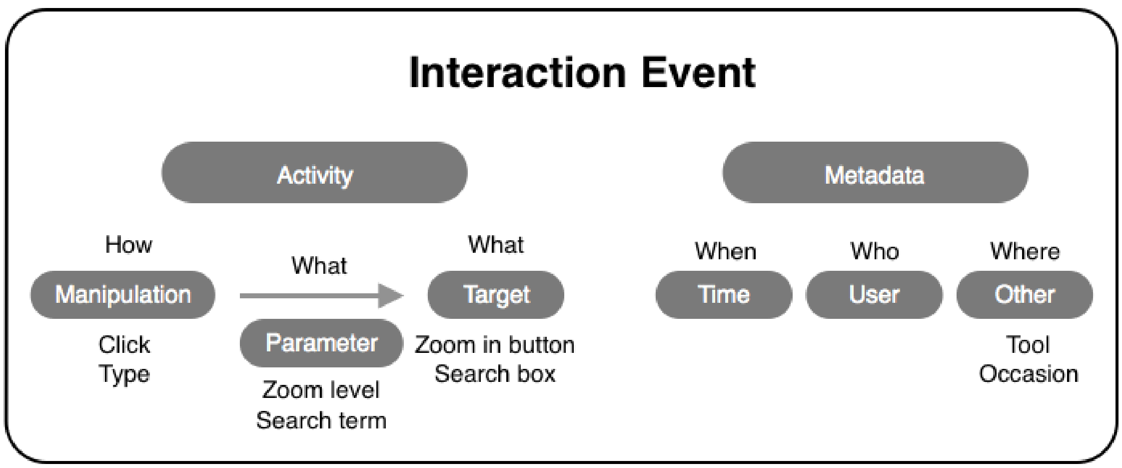


Figure 1: Interaction event contains two components: Activity and Metadata.

2.2.2 Analyst

An analyst interprets interaction events from usage logs. In visualization research, this person could be the same person that designed the application and logged the interaction events. As a result, I make the assumption that the analyst should be very familiar with the features of the application, details of the logs, and the log analysis goals. However, this may not be the case for large projects with software development teams. But even if an analyst only took the role of interpreting the logs, he/she should still need to learn about the design and interaction logging aspect of the application to conduct the analysis.

2.3 Analysis Techniques

There are three frequently used analysis techniques for extracting patterns from interaction events. These techniques could be used in isolation or in unison. Patterns extracted from these techniques are typically of different levels of complexity and can answer different research questions. Therefore, depending on the analysis goal, the best techniques for analyzing interactions may be different.

Hilbert and Redmiles surveyed techniques used for analyzing usage logs in HCI for studying usage and usability back in 2000 [20]. Many techniques for transformation, analysis, and visualization in this paper were related to the analysis techniques in this section. They found a general need of an analyst to interpret data extracted from automated computational methods. The authors also noted that visualizations can be useful for involving analysts in the process. However, the limitation then was that visualizations that were not simple charts mostly required an analyst to manually generate. Today, we can more easily create interactive visualization applications to support this process.

2.3.1 Categorize Events

This analysis technique organizes interaction events into new, semantically meaningful categories. Typically, a category is a composed or abstract representation of individual events. It can be derived from any attribute of an interaction event. For example, events can be categorized by time, user intent, or tool feature. Categories can also be custom defined by an analyst. An analyst decides how events should be categorized based on the analysis goal. In this section, I present the variety of techniques for categorizing events in the research literature.

Time

Events can be aggregated by time. Which time unit to use depends on the analysis goal. For example, Nicholas et al. grouped user interactions by day of week (e.g., Monday) when studying accesses to the online Blackwell Synergy digital library [30]. The interaction events were individual accesses to the online journals. The new aggregations represent user activities on the corresponding weekdays. This type of aggregation can be generalized into any time unit, such as seconds or years.

User Intent

Events can be mapped to user intent. A user intent can be seen as a low-level goal. The determination of user intent, unlike time, from interaction events may require some contextual knowledge from an analyst. For example, to determine a mouse click on a "zoom in" button, scrolling the mouse wheel up, and clicking on a "zoom in" menu item, all means a user intended to zoom into a view, requires a thorough understanding of how each interaction event is linked to the functions of an application.

To identify interaction patterns, many researchers used user intent to categorize events. For example, the interaction categories defined by Yi et al. used user intent to classify interactions [50]. They defined seven visualization-independent interaction categories: Select, Explore, Reconfigure, Encode, Abstract/Elaborate, Filter, and Connect. Pohl et al. used this categorization to analyze interactions from two different visualization applications (VisuExplore and Gravi++) [34]. They converted, for example, the "Diagram moved" interaction to "Reconfigure" for VisuExplore, and "Move (drag)" interaction to "Reconfigure" for Gravi++ (Figure 2). One notable observation with this categorization is that the interactions in the application were not evenly distributed into the categories. For example, VisuExplore had five out of thirteen interactions that mapped to the "Explore" category but Gravi++ only had one interaction that mapped to this category. However, if the new categories made sense to the analyst, the unbalanced assignments only further supported the understanding and comparison of the two applications. Similarly, Guo et al. also used Yi et al.'s categories as the basis for creating "high-level actions" for visualization evaluation [16]. In their work, they referred individually logged interaction events as "low-level actions" and can be categorized into "high-level actions." They used six of Yi et al.'s categories and added a new one, "retrieve," in their analysis.

Table 1 Interactions and corresponding categories of interaction [29]

Interaction	Categories of interaction
VisuExplore	
Vertical scroll bar moved	Explore(scroll)
Pan	Explore
Time scroll bar moved	Explore
Diagram resized	Encode
New diagram	Encode
Select diagram/data point	Select
Timemeasure tool result	Time.select
Diagram moved	Reconfigure
Closing diagram, deleting all diagrams	Reconfigure
Diagram collaps/expand	Reconfigure
Tooltip shown	Abstract/elaborate
Opening table panel	Abstract/elaborate
Zoom	Abstract/elaborate (zoom)
Gravi++	
Starglyph show	Encode
Attraction fields show	Encode
Traces show	Encode
Move (drag)	Reconfigure
Add	Filter
Remove	Filter
Time (time control)	Time.explore
Highlight	Select
Tooltip (hover)	Abstract/elaborate

Figure 2: Interaction event categorization by Pohl et al. using Yi et al.'s interaction categories [34, 50]. (Figure from [34]: Table 1)

Gotz and Zhou used a different "action" definition that was based on the Activity Theory and user intent for analyzing insight provenance [13]. They argued that converting interaction events into actions would provide an analyst with more semantically meaningful data during the analysis. An example of an action includes a type (e.g., pan), an intent (e.g., visual change), a list of parameters (e.g., range constraint list), and a description (e.g., a request to scroll a visualization to a new location along an ordinal dimension). Three types of actions were defined: exploration actions (e.g., filter), insight actions (e.g., annotate), and meta actions (e.g., edit), as shown in Figure 3. They also identified four types of user intents: data change, visual change, notes change, and history change.

Some studies use other interaction categories based on user intent. For example, Blascheck et al. [4] used abstract visualization tasks from Brehmer and Munzner's multi-level task topology [5] as their interaction categories for evaluating visualization applications. The task topology for the interactions include 11 tasks such as encode, select, and annotate. Nguyen et al. used search (keyword search, location search, route search, filtering) and reading (highlighting, annotation) activities as categories for studying the sensemaking process during web search [29]. Alspaugh et al. used 17 categories they created such as aggregate, augment, and cache, for analyzing the query log from a data analysis platform [1].

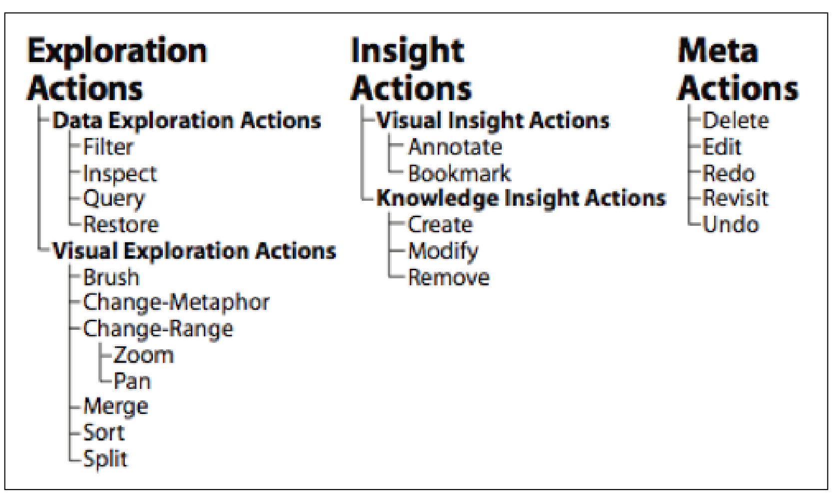


Figure 3: Type of actions defined by Gotz and Zhou. (Figure from [13]: Figure 2)

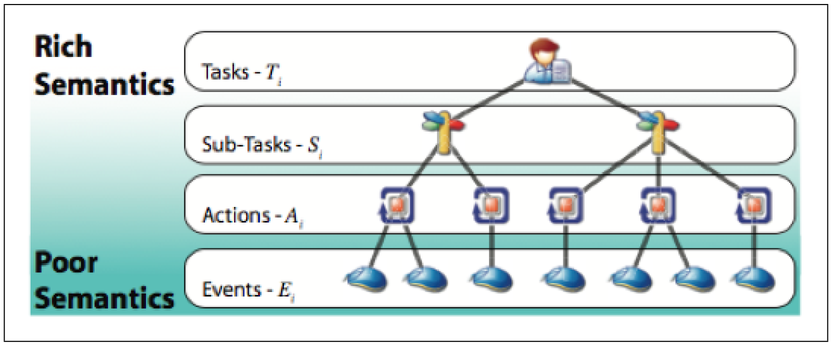


Figure 4: Gotz and Zhou defined multiple levels of user activities. The Action level is below the higher Task and Sub-Tasks levels. (Figure from [13]: Figure 1)

Tool Feature

Events can be mapped to tool feature. For example, a low-level tool feature can be "zoom in" so all interactions that zoom in to the view can be mapped to this category. On the other hand, a high-level tool feature can be a "line chart" so that all interactions in this chart can be mapped to one new category.

Benevenuto et al. mapped interactions into categories based on features to compare behaviors on different online social networking websites [3]. For example, they mapped "browse messages" and "write messages" interactions into "messages." The new categorization was used, for example, to compare feature usage frequency between websites (Figure 5). The categorization helped connect interactions between the compared websites and reduced the data variety to be manageable for analysis. Schneider et al. also analyzed social networking data to study similar interactive activities [37]. They created a similar set of feature-based categories from the interaction events, such as messaging and photos, in their analysis.

Other

Events can be categorized by other criteria custom defined by an analyst. For example, Card et al. created the category type called "problem space" to explore user activities with their Web Behavior Graphs (Figure 6) [8]. Each interaction was mapped

Comparison of popular user activities across four OSN sites.

Rank	Orkut		MySpace		LinkedIn		Hi5	
	Category	Share (%)	Category	Share (%)	Category	Share (%)	Category	Share (%)
1	Profile & Friends	41	Profile & Friends	88	Profile & Friends	51	Profile & Friends	67
2	Photos	31	Messages	5	Other (login)	42	Photos	18
3	Scrapbook	20	Photos	3	Messages	4	Comments	6
4	Communities	4	Other (login)	3	Search	2	Other (login)	4
5	Search	2	Communities	1	Communities	<1	Messages	3

Figure 5: Benevenuto et al. used feature-based categories to compare user activities between social networking websites. (Figure from [3]: Table 7)

to a problem space. For example, a user typing a URL was in the "URL problem space." Clicking links, images, and back were in the "Link problem space." Searching keywords was in the "Keyword problem space." Scrolling and eye-movement (from eye tracker) was in the "Visual search space." The Web Behavior Graph then connected the interaction events and mapped them to the problem spaces with different enclosing border types (e.g. dotted, solid). Similarly, to study user behavior and strategies, Reda et al. converted user interactions into two custom categories: ones that significantly changed the layout (e.g., creating, closing, and positioning views) and ones that did not (e.g., brushing-and-linking) [35]. Using this two-class categorization, they combined the new categories with information from verbal protocols to study visual analysis strategies. This type of custom creation provides the maximum flexibility in categorizing events.

Defining the appropriate categories can be a simple or challenging task. If the categories come from a predefined unit, such as the day of week, it can be straightforward to create them as there is no ambiguity in the categorization process. On the other hand, if the categories require an analyst to define them based on some contextual knowledge, it may be challenging to create. For example, the categories created by Yi et al. based on user intent were so complicated that they wrote a full paper about them [50]. Moreover, once the categories are determined, an analyst will still need to map interaction events to them. As a result, the categorization can be a laborious process.

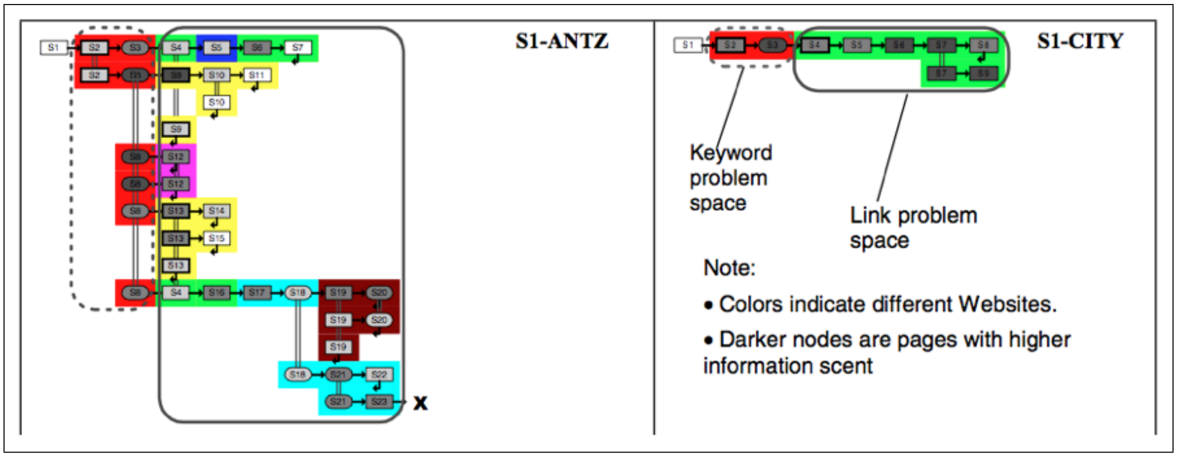


Figure 6: Card et al. designed the Web Behavior Graph that shows interaction events with custom categories (problem space) that used different enclosing border types (e.g. dotted, solid). (Figure from [8]: partial of Figure 4)

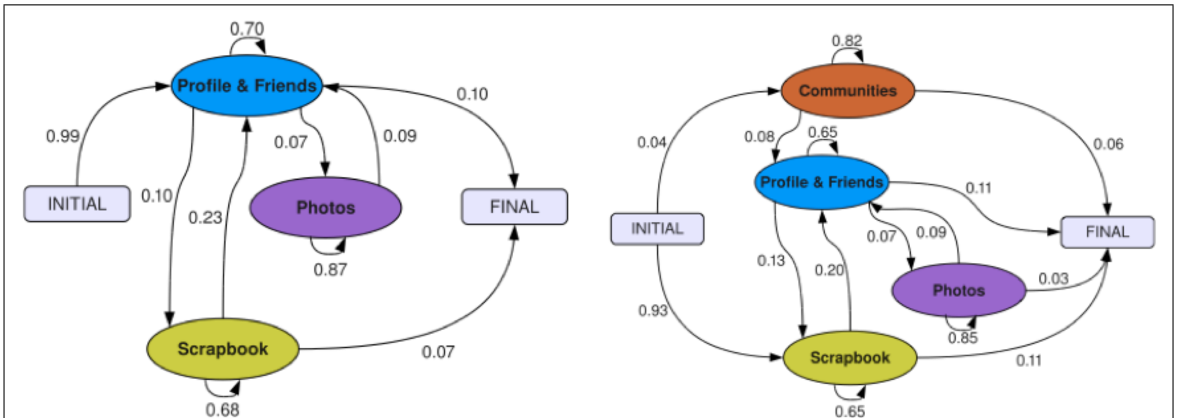
2.3.2 Extract Sequential Patterns

This analysis technique extracts sequential patterns of consecutive interactions. Sequential patterns of consecutive interactions can show higher-level patterns that include more than one event or category. For example, the sequence "open annotation dialog" → "click add annotation button" indicates a user added an annotation. These sequences can be manually defined by an analyst or automatically extracted from the log data.

Sequential patterns can be manually defined by an analyst. For example, Heer et al. used "chunking" to study interaction patterns for evaluating a visualization application [19]. They manually created a set of rules to group a variety of interaction sequences into a new analysis unit. For example, sequences of formatting interactions were chunked into the "formatting action." Similarly, sequence of sort and filter interactions that occurred less than 30 seconds apart were chunked into the "filter and sort action." Gotz and Wen manually identified a set of "patterns" from certain common sequences of "actions" (mentioned earlier in the "Category: User Intent" section (2.3.1) [13]) to detect "visual inertia (activity)" and recommend visualizations [14]. Patterns are, for example, a series of inspect actions (scan pattern). They defined four simple patterns: scan, flip, swap, drill-down, and used the identification of these activities as the basis to recommend visualizations to users. Similarly, Guo et al. defined "patterns" based on frequent sequences of "high-level actions" for visualization evaluation [16]. They identified four patterns: orienting, locating, sampling, and elaborating, that seem closely related to Gotz and Wen's patterns. For example, the "elaborating" pattern seems to be the same as the "scan" pattern that include a series of data inspection activities. Blascheck et al. designed a visual interaction analysis system that supports analysts to manually search for specific interaction patterns from a visualization application [4]. They have a visual interface in their analysis system for defining an exact or fuzzy search of a pattern of interest. The fuzzy search allows wild card parameters where an unspecified number of irrelevant items can be allowed within the search pattern.

On the other hand, sequential patterns can be automatically extracted. For example, Pohl et al. calculated the transition probabilities of visualization interaction categories to identify user strategies [34]; Coull et al., Iglesias et al., and Schonlau et al. used sequential patterns of user commands to identify users [11, 22, 38]; Weichbroth et al. extracted frequent sequences of webpages to study web navigation activities [48]; Hollink et al. studied the sequential patterns of online search to find higher-level user activities and interests [21]; Sinha et al. extracted frequent interaction sequences from online MOOC videos to classify user activities (e.g., rewatch, skipping) [42]; Brown et al. calculated short sequential patterns (2-3 events) of visual search interactions (e.g., zoom in, pan up) to classify users by their performances [6]. Blascheck et al.'s visual interaction analysis system can automatically calculate n-grams, all common substrings, and the longest common substring of event categories to help analysts identify interaction patterns for visualization evaluation [4].

As a sequential pattern is a number of interconnected interactions, it is typically visualized for analysis. The most common visualization is a node-link graph. The nodes, which represent interactions, are connected with edges that show the transitions between them. For example, both Benevenuto et al. and Schneider et al. used this type of transition graphs with website features (e.g., photos) as nodes to indicate the common usage patterns on social networking websites [3, 37]. Clustering sessions with similar activities, Benevenuto et al. presented what a “typical session” looks like with regards to the interaction transitions (Figure 7). Schneider et al. observed how website features connected with each other. They found for example, users most frequently leave the Home category for the Messaging category. Reda et al. used similar state transition graphs to show how visualization interactions could be combined with qualitative data from verbal protocols to examine user strategies [35]. Lam et al. included a state transition graph in the high-level aggregate pane in their application Session Viewer that visualizes web search interactions [28]; Pachidi et al. used a graph to show web access transition patterns [31]; Pitkow and Bharat used a graph to visualize website structure and webpage access statistics [33].



(a) Cluster 1 (b) Cluster 2

Figure 7: Benevenuto et al. showed transition graphs from the Markov models of “typical sessions.” (Figure from [3]: Figure 7)

Another type of visualization uses a transition matrix to visualize the graph data. Zaïane et al. visualized web access transitions with a transition matrix as well as a tree-based graph [51]. Zhao et al. also used transition matrices, but connected multiples of them, to investigate web navigation patterns [52] (Figure 11).

A third type of visualization uses a tree structure to show sequential patterns. For example, Card et al. created the Web Behavior Graph that visualizes web browsing interactions with a graph visualization using a tree layout [8]. Spiliopoulou and Pohle used a tree structure to visualize webpage access sequences [43]; Wongsuphasawat and Lin, and Shen et al. both used a icicle tree visualization to show frequent web interaction sequences [39, 49]. Wongsuphasawat and Lin used it for analyzing Twitter activities (Figure 9) and Shen et al. for eBay interactions.

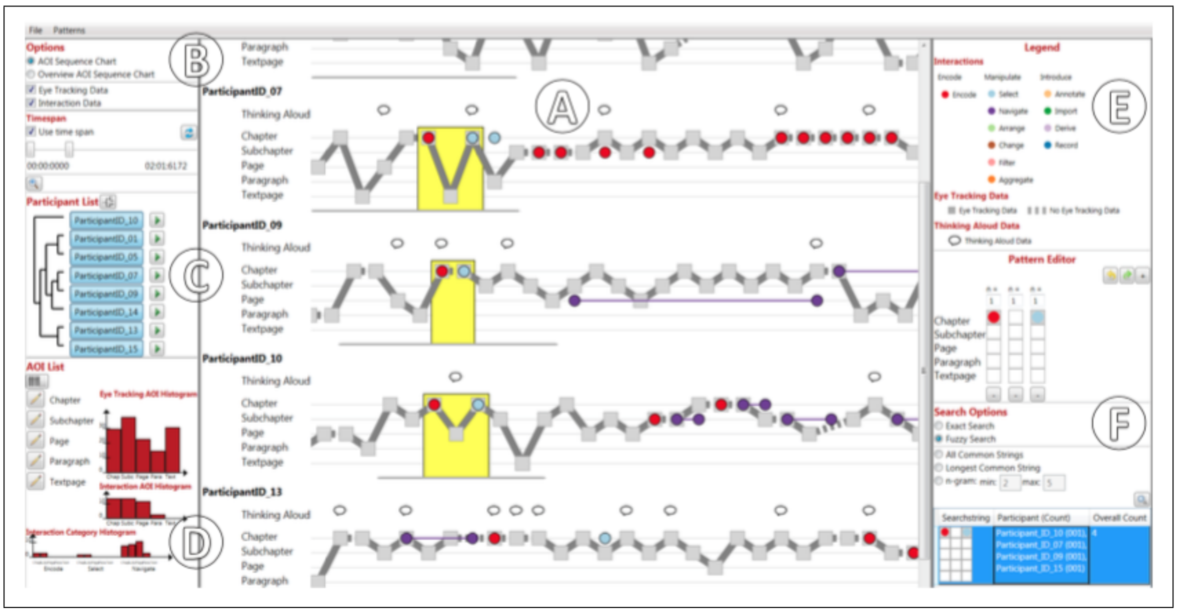


Figure 8: Sequential pattern highlighted within original interaction sequences [4]. (Figure from [4]: Figure 1)

Yet another type of visualization uses a timeline layout. Blascheck et al. visualized interaction sequences on a timeline and highlighted sequential patterns, as shown in Figure 8. This visualization is less scalable but preserves the temporal context of the sequential patterns [4]. Nguyen et al. designed SensePath that also visualizes interaction sequences over time [29]. However, the analysis system does not explicitly support the identification of specific interaction sequences. Analysts need to identify patterns visually by themselves.

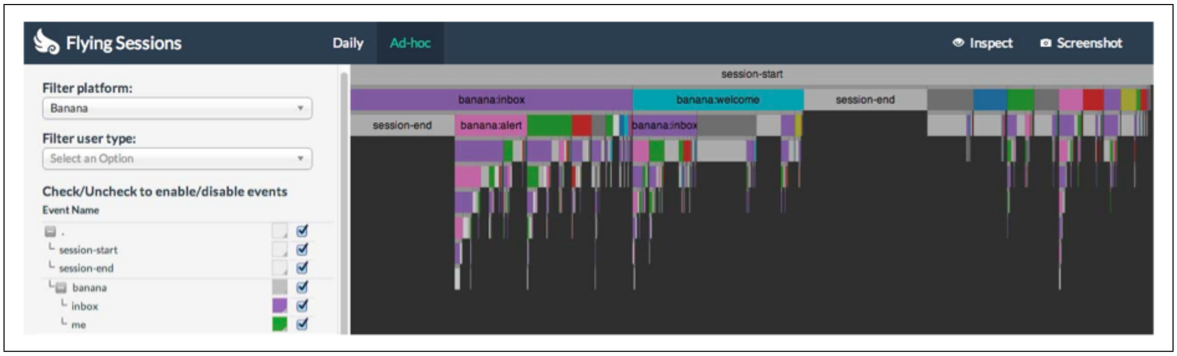


Figure 9: Wongsuphasawat and Lin used icicle tree to show sequential patterns and their frequency distribution (Figure from [49]: Figure 8)

Wei et al. took a different approach in visualizing frequent web interaction sequences as shown in Figure 10 [47]. They visualized each frequent interaction sequence as horizontally-connected color blocks. Each color block represents an interaction. Then, they sized the sequences based on their occurrence frequency and arranged them on a view with some criteria such as keeping similar sequences nearby and avoiding overlaps. The resulting view showed the most popular sequences in the center of the view with similar ones clustered together.

Frequent sequential patterns of consecutive interactions can be automatically extracted so they are easy to obtain. However, they can be difficult to meaningfully interpret. For example, what does a sequence of "print → zoom → filter" mean? The knowledge of an analyst is as a result important in figuring out which sequences are meaningful and informative to extract. Moreover, finding these patterns alone may not help identify a user's high-level analysis method because a frequent sequential pattern may only cover a small portion of the usage session. For example, knowing that users have used many "scan" patterns does not inform about what frequently led to the scanning and what resulted from the scanning. In other words, picking out individual patterns can miss out the broader context of the entire interaction session to know for example, the items being scanned are often from a search result, that might better help an analyst understand a user's overall analysis method.

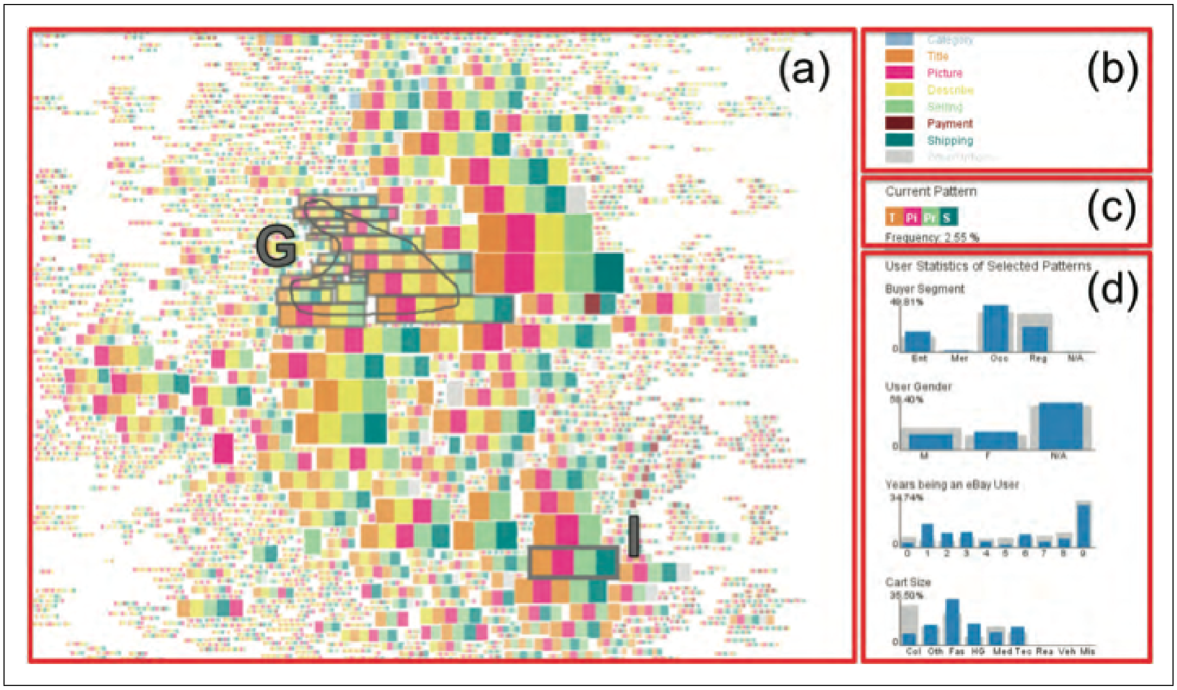


Figure 10: Wei et al. used clustered colored blocks to show sequential patterns and frequency distributions (Figure from [47]: Figure 6)

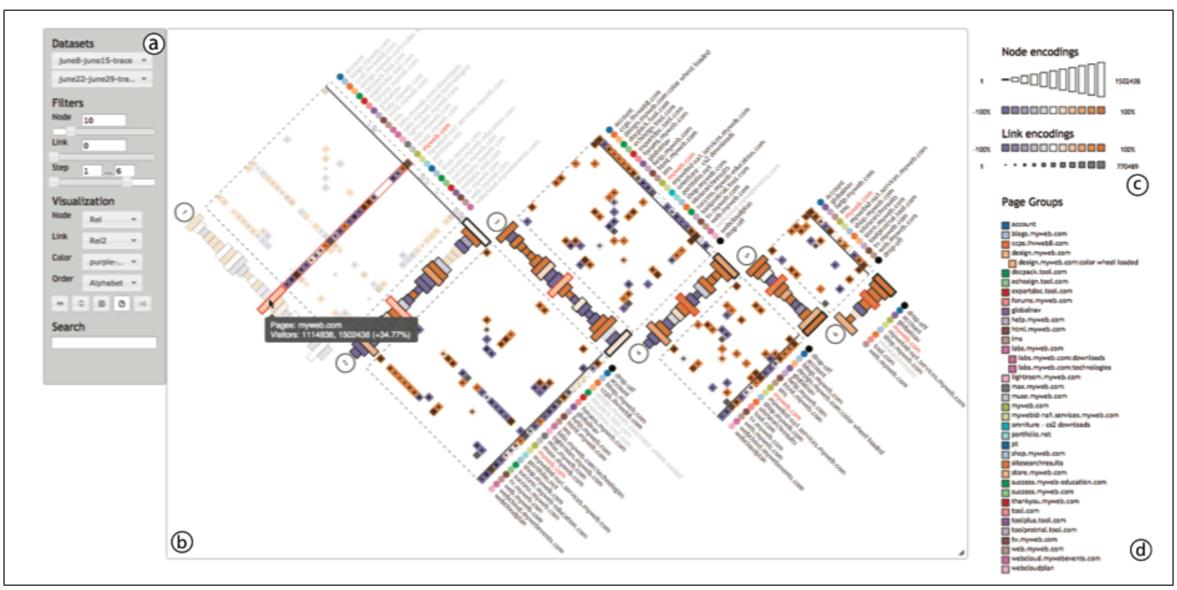


Figure 11: Zhao et al. designed MatrixWave with connected transition matrices to explore web navigation patterns (Figure from [52]: Figure 5)

2.3.3 Generate Frequency Distributions

An analyst generates a frequency distribution of events, categories, or sequences over certain contextual information to identify a high-level, overall usage pattern. For example, the frequency distribution of interaction events over tool features can help determine which features may require more interactions to operate.

Frequency distributions can be presented in a table. For example, Pohl et al. distributed transition probabilities, which reflected the occurrence frequencies, of visualization interactions over the corresponding transitions in a table to show the most common user strategies [34]; Catledge and Pitkow distributed frequencies of web interactions in a table to show their relative prevalence [9].

A more common visual representation of frequency distribution is with a chart. For example, one typical chart for these data is a histogram. Nicholas et al. distributed interaction frequency over day of week to compare weekday to weekday online journal access activities [30]; Iglesias et al. distributed user command frequency over command sequences in a histogram to characterize user activities [22]; Pepyne et al. visualized frequency of interactions on a chart with time on both axes (x-hour, y-day) to profile regular users with their active/idle time pattern [32]. From modeling behaviors of regular users, they can detect intrusion from other users; Catledge and Pitkow charted web interaction frequency over web navigation path length to characterize web browsing strategies [9]; Kim et al. and Chorianopoulos both visualized video interaction peaks, which are based on interaction frequency over time, in line charts to predict users' interests in videos over time [10, 26]; Also analyzing video interactions, Shi et al. used a stacked area chart to show interaction frequency over time for understanding MOOC video viewing activities [40].

Many of the more advanced visualizations used for showing sequential patterns earlier can also encode the frequency distribution of those patterns. For example, the state transition graph that shows transition probabilities of sequential patterns could encode the frequency distributions of transitions on the edges [3, 35, 37]. Similarly, the alternative visual representation of the transition graph, the transition matrix, could encode the frequency distribution of transitions in its cells [51, 52]; Yet another related visual representation of sequential patterns, the icicle tree, could encode the frequent distribution of sequences with the size of the corresponding item in the tree [39, 49]; Wei et al.'s method of encoding the frequency of sequential patterns with the size of the corresponding colored blocks was a different way of showing the frequency distribution of interaction sequences [47].

For interactions on a UI, there is typically a spatial mapping of where an interaction occurred. Therefore, an alternative and more intuitive way to visualize interaction frequencies is to visually encode this information on the UI layout they were acted on. Gray et al. used this method about 20 years ago to visualize interaction frequencies ([15], see Figure 12). This method can help an analyst at a glance understand the frequency distribution of interactions on a UI layout. The challenge with this method in visualizing visualization interactions is that today's user interactions are much more complex. For example, some interactions, such as pressing a keyboard shortcut, may not have a natural spatial mapping to a UI component. Moreover, the same spatial locations on the screen may accept different types of interactions, such as left- and right-clicking the mouse buttons, for different actions. Last, visualization applications can have multiple views, numerous dialogs, and many menu items with UI elements changing in different modes. Attempting to show all the possible views that a user saw to encode the frequency distribution this way could take a significant amount of screen space and reduce the efficiency of visual analysis. Therefore, using this technique for visualization interaction presents many challenges.

Jeong et al. and Dou et al. took a different visualization approach by mapping the frequency distribution of interactions onto multiple treemaps to find user strategies in financial visual analysis [12, 24]. They designed a visual interaction analysis system

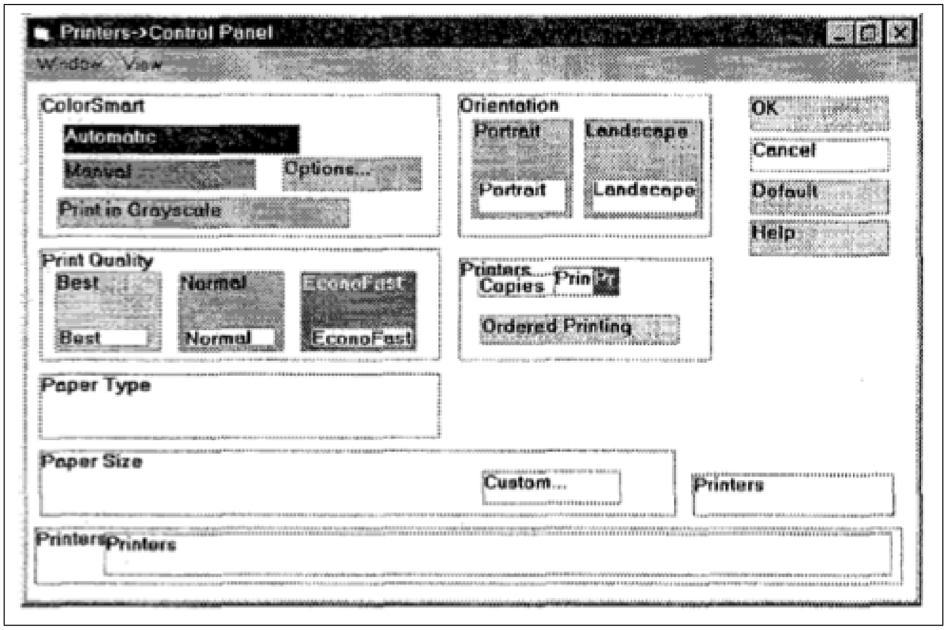


Figure 12: Gray et al. visualized the frequency distribution of UI elements on the original UI layout. (Figure from [15]: Figure 3)

called the "strategy analysis tool" specifically for this purpose. The system shows a set of treemaps that groups the interactions by different data dimensions such as time, keyword, and account (Figure 13). Depending on the clustering of interactions, they can identify different strategies, such as users searching for specific keyword combinations. Using multiple treemaps to map interaction frequencies to different data dimensions provides a more flexible and space-efficient mapping of frequency distribution to the visual layout than mapping them to the actual UI layout. The tradeoff is that the spatial locations are assigned new meanings that need to be learned by an analyst.

As shown above, frequency distributions of activities over a certain contextual variable can be used to identify a high-level, overall pattern. Visualizations are particularly useful for showing frequency distributions either over time, space, or other contextual variables. However, any event, category, or sequential pattern can be counted and distributed over any contextual variable. Which combinations to select to identify patterns can be a challenge to an analyst. Moreover, numerous visual

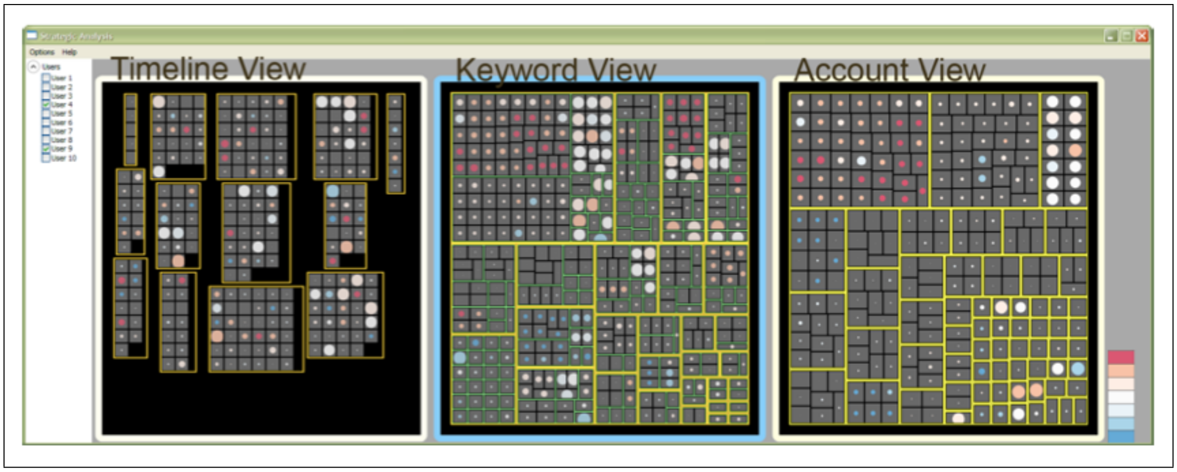


Figure 13: Jeong et al. designed the strategy analysis tool to visually identify user strategies. (Figure from [24]: Figure 5)

representations of frequency distributions are available. Each of them is preferable in some way at showing one aspect of the data. As a result, how to select the appropriate distributions and visualizing them are two key challenges with this analysis technique.

2.4 Limitations

As discussed in this chapter, each technique for identifying activities has its limitations. Finding a variety of patterns with different levels of composition and abstraction can be ineffective and labor-intensive without the proper tools and guidelines. These issues are mostly because the identification of these patterns often still require an analyst's subjective assessment and interpretation. Because of the need for analyst involvement, the entire process suggests a visual analytics solution. As discussed, a few research projects already used visualization techniques to support such analysis with varying levels of success. However, many of these techniques were only designed to support the identification of certain types of patterns for those specific projects. Some researchers used these techniques together but each has their own way of implementing the integration. Without an established method and framework for systematically extracting patterns from interaction data, visualization researchers typically do not even consider analyzing interaction logs in depth. I believe this is a significantly missed opportunity in the visualization research community because of the vast amount of information about users' visual analysis process that could be discovered at scale.

CHAPTER III

VISUAL INTERACTION ANALYSIS

To establish an interaction analysis method and framework, I first present a list of tasks, or goals, of an interaction analyst based on the literature review and my own analysis experience. Next, I present an interaction analysis method that systematically applies a set of analysis techniques. Last, I present a visual interaction analysis framework that utilizes visual analytics to implement the method.

3.1 Interaction Analysis Tasks

I examined typical interaction analysis tasks from the literature and my analysis experience [18]. These tasks were not exhaustively extracted from a systematic review of the literature but they should address a large portion of typical analysis needs.

3.1.1 Assess Usability

An analyst seeks to examine how easy it is to learn and use a visualization application. This information could be gleaned from the interaction log in many ways. For example, a frequently used feature should be easy to use. Similarly, if a feature is picked up quickly by users, it should be easy to learn. But these simple metrics may not always be reliable because a feature can be difficult to use but supports a task that no other tool does. Therefore, users had no choice but to use it. Also, how "quick" is quick enough for picking up a feature to show it is easy to learn? As a result, having a good understanding of the context of use, such as the availability and learnability of feasible alternatives, can help more reliably interpret the interaction data.

A more advanced method to assess the learning curve and the usefulness of features in a visualization application is to measure the change in its users' interaction patterns. For example, a hypothesis is that when a user is less familiar with an application, his/her interaction patterns are random, inefficient, and experimental. But as the user learns about which features work better, his/her interaction patterns become consistent, efficient, and predictable. Therefore, by identifying this interaction pattern change, an analyst can assess the learning curve of an application as well as which features might be more useful than others.

A general metric for assessing the efficiency of an application is to examine the amount of user interactions, such as mouse clicks, required to accomplish a task. Typically, fewer interactions are considered a desired outcome. However, for exploratory data analysis in visualization tools, this outcome may not be preferred because more interactions may lead to different views and a better understanding of the data. In other words, this metric may not apply well anymore.

3.1.2 Assess Utility

Card et al. declared that "the purpose of visualization is insight, not pictures" [7]. Therefore, an analyst seeks to examine whether a visualization application is able to help discover insights to determine its utility. Typically, insights are measured from interviewing users because it is a subjective assessment of a finding. An insight to one user may be common sense to another. However, interviewing users is labor-intensive. Nowadays, users may not even be approachable to an analyst because many visualization applications are deployed online. Therefore, the question is whether it possible to identify moments of insights merely from a user's interactions? Gotz and Zhou provided a potential answer to this question by identifying a set of "Insight Actions" that may be connected to insights. For example, "Bookmark" and "Annotate" activities are considered "Visual Insight Actions" because one reason people bookmark or annotate views is when insights are discovered.

However, not all bookmarking and annotating activities indicate an insight. For example, a user may use an annotation to simply add information to a chart. As a result, to identify bookmarks and annotations that do indicate insights, an analyst needs to manually examine the content of the bookmark or annotation. For example, if a bookmark is labeled "Discovered unexpected surge in...," it clearly indicates an insight was found. On the other hand, if a bookmark is labeled "Saved 10/1," an insight may or may not have been found. This example also illustrates the challenge in interpreting the content of bookmarks and annotations written by users. As a result, in order to be certain about insight discoveries, an analyst can ask the users to indicate them in the logged interactions. For example, when a user saves a bookmark, perhaps there is an easy way for the user to tag whether it is because of an insight. This way during the analysis, insights could be automatically extracted based on these tags so that an analyst will not need to unreliably examine each visual insight action to determine whether an insight was discovered nor need to interview users about them.

3.1.3 Learn About Users

An analyst seeks to learn about users of visualization applications from analyzing interaction logs. For example, which users are expert visualization users and which users need some additional directions based on their interaction patterns? Sometimes, a feature in an application can be optionally operated from different UIs. For instance, users may be able to zoom into a view using either a zoom-in button or a zoom-in menu item. Which method is preferred by the users? A visualization may be used for multiple usage occasions, such as for a presentation or for monitoring real-time events. How do users interact with an application differently under different occasions? At an abstract level, an analyst seeks to identify users that exhibit certain behaviors (e.g., expert analysis) or identify user behaviors under certain circumstances (e.g., presentation). Learning about users is an important step towards designing a more effective and usable visualization application.

3.1.4 Understand Usage Patterns/Analysis Methods

An analyst seeks to explore the variety of ways a visualization application can be used. In particular, frequent usage patterns are valuable to discover because they reveal which features are more useful in actual use cases. Some patterns are expected by an analyst. For example, one can expect users to be adjusting the zoom level in a map-based visualization. But which other unexpected patterns might there be? Using observations and interaction log analysis, Kang et al. [25] discovered a variety of usage patterns in Jigsaw [44], a visual text analysis system. For example, one pattern started from users scanning the text documents, filtering down to a subset of interest, and then reading the subset of documents in detail. Another pattern started from repeated searches of different keywords and then reading documents in the search results. Some of these patterns might not have been expected by an analyst so they provided more insights to the varying ways this type of visualization applications were operated in practice.

One or more usage patterns may indicate a specific visual analysis method (VAM) is taken. A visual analysis method is a methodological and semantically meaningful approach in using a visualization application. A VAM has been called a visual analysis "strategy" [12, 25, 34, 35], "method" [12], "interaction model [46]," or even a "mantra" [41]. Many researchers seek VAMs because they show interpretable usage patterns that can help derive a deeper understanding of how a visualization application is used at a high level. One well-known example of a VAM is Shneiderman's visual information-seeking mantra [41]. The mantra, "Overview first, zoom and filter, then details on demand," represents a high-level, semantically meaningful method for operating a visualization application. It can be discovered from a usage pattern that begins by displaying all the data in the view to create an overview. Next, the pattern should be followed by a set of zooming and filtering activities. The zooming and filtering could occur in any temporal order and combination. Afterwards, the pattern ends with an action that extracts more details from the dataset. This VAM is widely used and can exist in any type of visualization application. On the other hand, other VAMs can only exist in a certain type of visualizations or data types. For example, the VAMs, or strategies as the authors call them, discovered in Jigsaw by Kang et al. were specifically for visually analyzing text documents with a specific set of visualization techniques supported in Jigsaw [25]. This type of VAMs may not be generalizable to other visualization applications but they are particularly useful for examining the unique usage patterns in a specific type of application.

3.2 Interaction Analysis Method

From the analysis tasks, it is apparent that an analyst needs to not only identify usage patterns of individual interaction events but also examine those from groups of events in categories and sequences. But how should these categories and sequences be identified from events? In this section, I present an interaction analysis method for extracting a hierarchical set of semantically meaningful patterns from interactions. These patterns are able to more directly support the analysis tasks of an analyst.

3.2.1 Step 1: Log Events

The first step is to identify events by logging them from visualization applications (Figure 14, step 1). An event is an interaction such as clicking the Zoom In button. It is the basic unit collected in the interaction log and is the same as the event in the lowest-tier of user analytic activity defined in Gotz and Zhou [13]. A logged event should contain both the interaction activity (e.g., click Zoom In button) and context (e.g., time, user) as defined earlier in Section 2.2.1.

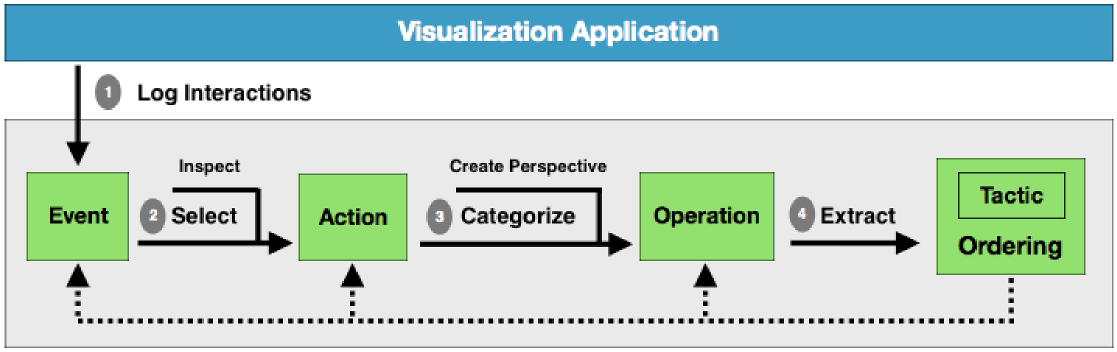


Figure 14: Interaction analysis method. (1) Log interaction events from a visualization tool (2) Inspect and select events as actions (3) Categorize actions as operations with an analysis perspective (4) Extract tactics and orderings from frequent operation sequences.

To identify all potential events to log from a given visualization application, an analyst needs to traverse all UI components that users may interact with. For each UI component, the analyst should decide whether the interaction is useful to log and if so, how much information should be logged to unambiguously identify it. For example, clicking the Zoom In button can be unambiguously logged as an event if there is only one Zoom In button in the application. Moreover, this event maps directly to an unambiguous function: zooming in the view. However, if an interaction such as clicking on a view may function differently when the view is in a different mode, then it should not be logged as an event. In this case, additional contextual information needs to be logged with the interaction activity to turn it into an unambiguous event. For example, if the different modes of the view are determined by whether data are shown in the view, the events should be (1) clicking on the view "when no data are shown" and (2) clicking on the view "when data are shown." My criteria for defining an event as a result is stricter than simply defining every interaction on the UI as an event. Each event needs to include enough contextual information to be independently interpretable without referencing any other surrounding events. This level of event logging, though more difficult to achieve, could provide more useful information for the analysis.

3.2.2 Step 2: Inspect and Select Events as Actions

After events are collected, an analyst needs to select the ones that are relevant to the analysis (Figure 14, step 2). Irrelevant events are common in the data because at the time of logging the events, an analyst typically takes a "log as much as possible" approach. These events, if not filtered, becomes noise in the data that can overshadow otherwise obvious patterns. To determine which events are relevant, an analyst needs to inspect the events. This inspection can also remind an analyst about the logged events, assess the quality of the logged events, and determine analysis goals. Sometimes an analyst would prefer to select multiple consecutive events as a single unit. For example, visualization applications typically provide more information about a data item when a user moves the mouse cursor over a glyph representing that data item. When the mouse cursor is moved away from the data item's glyph, the information disappears. These two events, mouseover and mouseout a data item's glyph, typically occur in this specific sequence and together represent a single user action. Therefore, an analyst may choose to compose and select these events together into a new unit in this step, as shown in Figure 15a. I call these selected events and composed event groups "actions."

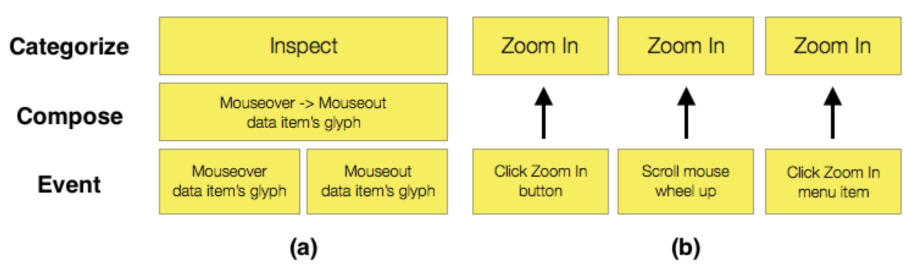


Figure 15: Illustration of the difference between composing and categorizing events. (a) Both compose and categorize events. (b) Only categorize events.

My definition of action is similar to that defined by Gotz and Zhou [13] but is broader in scope. They identified a fixed set of mid-level, abstracted actions, such as

Filter, Inspect, Brush, based on user intent. I purposely only defined a "method" for identifying actions but not a list of available actions. With this type of definition, an analyst can be more flexible in identifying actions at different levels. For example, an analyst can choose to identify lower-level individual user interaction events, such as "Click Zoom In button," as actions. This type of action is useful for the analysis but is not available in Gotz and Zhou's definition. Because actions are selected individual or group events, they will not overlap in time.

3.2.3 Step 3: Create Perspective and Categorize Actions into Operations

After actions are selected, an analyst categorizes them into "operations" (Figure 14, step 2). But how should actions be categorized? One way is to use an analysis perspective. An analysis perspective defines a set of operations based on some criterion. The simplest perspective is based on the actions identified in the previous step. It includes the full set of actions so that each of them can be individually mapped to a unique operation. This perspective allows an analyst to study each action separately.

A very common analysis perspective is based on user intent. This type of perspective includes a set of operations that each represents a unique user intent. For example, a Zoom In operation indicates an intent of zooming into a view. This intent could be fulfilled by more than one interaction event, such as "Click Zoom In button," "Scroll mouse wheel up," or "Click Zoom In menu item." When these actions are categorized into the "Zoom In" operation because they fulfill the same user intent, all these actions going forward will be recognized as the same "Zoom In" operation. Therefore, as shown in Figure 15b, the number of operations generated by the actions are the same as the original number of actions. An analyst could create this type of perspective himself/herself or use an existing task taxonomy. A task taxonomy includes a set of user tasks that are also based on user intent. For example, the interaction taxonomy developed by Yi et al. [50], which includes tasks such as “select” and ”filter,” was used by both Pohl et al. [34] and Guo et al. [16] in their analyses. Alternatively, an analyst can create perspectives based on other criteria. For example, Kang et al. created a perspective based on “views” in the visual text analysis tool, Jigsaw, to categorize actions [25]. Using this perspective, they were able to study the usage distribution of “views” from the interactions to identify analysis strategies. In general, if an analyst has a specific set of activities in mind that he/she cares about, the analyst should be able to create a perspective that can organize a set of actions about these activities into operations. Notice that not all actions need to be categorized into operations in a perspective. Actions that are left out of the perspective will not be further analyzed and are effectively filtered out from the analysis. Because different perspectives can include a vastly different set of operations, an analyst should carefully define perspectives based on the analysis goals.

3.2.4 Step 4: Extract Frequent Operation Sequences as Tactics and Orderings

After identifying operations, an analyst next extracts frequent sequential patterns of operations as “tactics” and “orderings” (Figure 14, step 4). A tactic is a sequence of two or more operations that occurred consecutively and over an analyst-defined frequency in the interaction log. Its level is similar to the “patterns” found in Gotz and Wen’s work [14]. Gotz and Wen manually defined a set of patterns that are based on sequences of consecutive interactions to identify a visual analysis behavior (visual inertia). For example, they defined the “scan pattern” by a series of data inspection actions. Guo et al. also identified four patterns in their work in a similar way [16]. As opposed to “patterns,” which are eventually manually defined by an analyst based on frequent sequences of consecutive actions, the identification of tactics in this model is completely data-driven. Therefore, tactics may not semantically map to anything meaningful. For example, “Zoom in → Pan” (Zoom in and Pan) is a tactic that shows an exploratory behavior. Finding this tactic indicates that the user may have used the tool for data exploration. On the other hand, “Print → Zoom → Filter” may not have such clear mappings. Nevertheless, both tactics, if emerged from data, show how users frequently interact with the visualization application. Two tactics may overlap when a subset of operations are part of both of them. This overlap could be due to transitions between tactics, such as when a user changes his/her tactic from “Zoom in → Pan” into “Pan → Inspect.”

On the other hand, orderings are sequences of operations that are not necessarily consecutive nor frequent in an interaction log. Tactics are a subset of orderings. Typically, only tactics are analyzed in studying interaction patterns [14, 16, 19, 34]. However, many useful sequential patterns may not require operations to occur back to back nor frequently in the dataset. For example, the use of the Visual Information-Seeking Mantra requires the identification of a sequence of “Overview,” “Zoom,” “Filter,” and “Show detail” interactions that occur in this order [41]. These interactions do not necessarily need to occur back to back. For example, if a user chose to change the font family of the texts after the overview and before a zoom, it does not matter because the identification of the mantra is not influenced by the occurrence of other interactions. This type of sequence is particularly useful for identifying high-level analysis methods such as the mantra.

3.2.5 Summary

In this section, I presented an interaction analysis method that step by step identifies actions, operations, tactics, and orderings from interaction events. A summary of these patterns extracted from the event organization process is presented in Table 1. However, how should this method be practically implemented? In the next section, I present a visual interaction analysis framework to address this challenge.

Table 1: Definition and example of patterns extracted from the event organization process.

Level	Definition	Example
Ordering	A sequence of operations that do not necessarily occur back to back	Zoom...Pan
Tactic	A sequence of consecutive operations that occur frequently	Zoom and Pan
Operation	A set of categorized actions	Zoom
Action	One or more selected (consecutive) events	Zoom In
Event	A low-level user interaction	Click Zoom in button

3.3 *Visual Interaction Analysis Framework*

A few steps in the interaction analysis method can be labor-intensive and be perceived impractical. For example, finding which events to select as actions requires the inspection of all the different type of events. Similarly, determining how actions should be categorized into operations requires an assessment of all the actions. The labor-intensive part seems to be mostly related to visual inspections. As a result, a potential solution to the problem is to increase the efficiency of the visual inspection process.

Two methods can increase the efficiency of the visual inspection process. The first method is to design a better visual representation that allows an analyst to more quickly uncover patterns of varying levels. The second method is to supply computational algorithms that can preliminarily analyze the data to provide suggestions to an analyst to reduce the need to manually inspect a large amount of data. These two methods are the key components of a visual analytics system. As a result, I employed visual analytics to increase the practicality of the analysis process with a visual interaction analysis framework as shown in Figure 16.

The framework is designed from the interaction analysis method. The first step

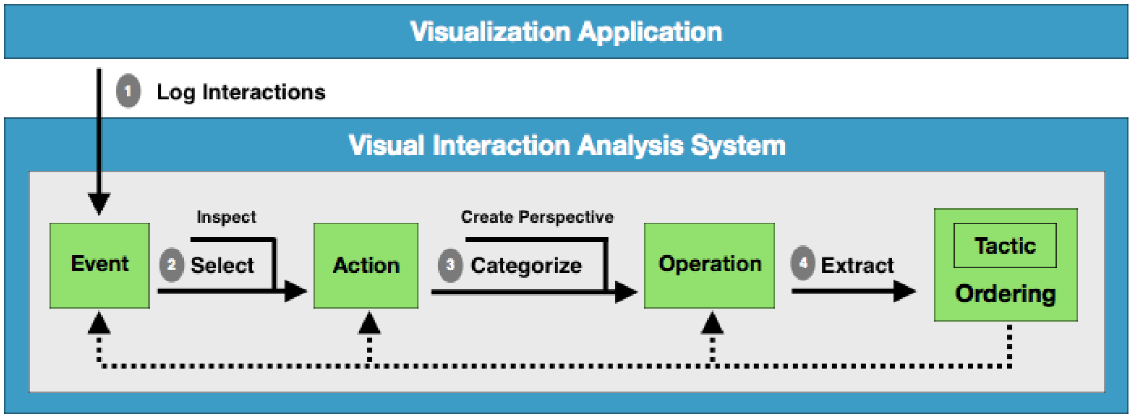


Figure 16: The visual interaction analysis framework for identifying patterns from interaction logs. (1) A visualization application automatically logs interaction events defined by an analyst. The logged interaction events are imported into a visual interaction analysis system. (2) Select events as actions after inspection. (3) Categorize actions as operations based on an analysis perspective. (4) Extract frequent sequences of operations as tactics and orderings.

of the framework is to log interaction events (Figure 16, step 1). An analyst defines which interaction events to log and the visualization application handles the actual logging. After the logs are collected, the interaction data are imported into a visual interaction analysis system as "events." The analyst next uses the system to inspect these events and select one or more of them as actions (Figure 16, step 2). Afterwards, the analyst defines an analysis perspective to categorize the actions into operations. (Figure 16, step 3). Frequent sequences of consecutive operations are next extracted as tactics and sequences of operations that do not not necessarily occur consecutively nor frequently are identified as orderings (Figure 16, step 4). This step will be led by the analyst and supported by computational algorithms such as sequential pattern mining algorithms with visual analytics. In visualization research, it is not uncommon for a researcher to be the analyst and handle all the steps. However, it is possible in practice that a designer built a visualization application, an analyst instrumented the application to log interactions, and another analyst explored the interaction log.

In this chapter, I provided a list of interaction analysis tasks and a method for identifying patterns to help achieve these tasks from interaction logs. To ensure flexibility and practicality in the analysis process, I employed visual analytics in a framework that implements the process. In the next chapter, I will present an implementation of this framework from the perspective of the most important component in it—the visual interaction analysis system.

CHAPTER IV

INTIVISOR: VISUAL INTERACTION ANALYSIS SYSTEM

I designed a visual interaction analysis system, IntiVisor, to implement the visual interaction analysis framework described in Section 3.3. In this chapter, I first present the design objectives of this system. Next, I discuss how interaction events could be logged as input to the analysis system. Afterwards, I present the visualization design of IntiVisor.

4.1 Design Objectives

IntiVisor is designed for the interaction analysis tasks described in Section 3.1 [18]. To support these tasks, I present the following design objectives.

D1. Support Event Organization

The system needs to help an analyst organize events into actions, operations, tactics, and orderings. Any feature that may lead to more flexibility in identifying different types of activities in the visual analysis process should be considered.

D2. Include Automated Computational Assistance

The system needs to help an analyst recognize patterns at scale through automated pattern detection algorithms. Automated algorithms are best at identifying potentially meaningful patterns that provide an analyst more information about the data in the analysis process.

D3. Provide Configurable Visualizations

The system needs to help an analyst visually and interactively inspect the data to support flexible event organization and exploration. Therefore, the visualizations should be easily configurable by an analyst to best reveal hidden patterns in the data.

D4. Apply to Any Visualization Application

As a generic analysis tool, the system should to be applicable to any visualization application. An analysis system is less useful if only interactions from a certain type of visualization application can be analyzed.

D1 is the design objective aiming to ensure the flexibility of the analysis framework, D2 and D3 are for increasing the practicality of the framework, and D4 is for the generalizability of the framework. I will next discuss the first step in the framework, logging interaction events.

4.2 Log Events

The input of IntiVisor are the logged interaction events. They are collected in the first step of the framework (Figure 16, step 1, on page 35). But which events and how much information about these events should be collected? As a general guideline, an analyst should keep as much potentially useful information about the users' interactions as possible. In this section, I will discuss the practical aspects of event logging in this framework.

An analyst first needs to find all potential interactions in a visualization application whose events would be logged. One simple method is to visually inspect all the UI components one by one to identify all the corresponding user interactions that could be logged. However, this task may be tedious and challenging because there may be many UI components and not every user interaction may be clearly mapped to a UI component. An alternative is to look for interaction event "handlers" in the code of a visualization application. For example, if the application is implemented in Java, an analyst could search for all the classes that implement the EventListener interface, such as those implementing the ActionListener. Each ActionListener object has an actionPerformed method that defines what happens upon a user interaction. Therefore, finding all these EventListener implementations can help find most of the user interactions. Similarly, if a visualization application is implemented in Javascript, the analyst could find user interactions from the Javascript event handlers that handle events such as "click" and "mouseover." But sometimes events of interest may not have event handlers in the source code of a visualization application. For example, scrolling the view in a window is a commonly used interaction. However, typically the default view-panning function is sufficient so the application designer does not need to manually create a custom event handler. In this case, if the scrolling interaction event should be logged, an analyst should add an event handler to log this event. Note that not all events need to be logged. For example, in Javascript, the "mousemove" events occur when the mouse moves. Logging these events can typically generate a significant amount of log entries. If an analyst does not care about these events and wishes to save some computational power in filtering these events out during the analysis phase, these events do not need to be logged.

After determining which events to log, an analyst needs to determine at which level of detail should the events be logged. For each event, both the metadata and activity attributes that were described in Section 2.2.1 should be logged. Conceptually, each attribute of an interaction event can be logged in a separate field as shown in Table 2. Activity attributes (manipulation, parameter, target) are the most important components for identifying a user's activity. Using three separate fields allows an analyst to be able to more readily separate the manipulation, target, and parameter, during the analysis. For example, the analyst could use a combination of the manipulation and target fields to identify the user's activity (e.g., "click" "zoom in button") regardless of the parameter (e.g, zoom level). Metadata attributes (time, user, other) provide context to the activity. The "other" attribute is particularly flexible to define any other contextual information that is useful to identify the event. For example, the "other" attribute can include a session identifier for distinguishing sessions. Alternatively, the "other" attribute can be the version of a visualization application or usage occasion. Conceptually, I only show one field for this attribute in the table. However, in an actual implementation, multiple "other" contextual variables can be placed in multiple fields in the table. At the every least, a logged interaction event should include enough detail to be independently interpretable. Therefore, the minimal number of metadata attributes to log depends on how many are required to uniquely and independently identify every interaction event.

Table 2: An interaction event can be logged in this format. Each field maps to an attribute in the interaction.

Manipulation	Parameter	Target	Time	User	Other
click	zoom level 2	zoom in button	01-01-2015 00:00:00	1	session

After the events to log are defined, an analyst next decides how to log them. This process can be as simple as directly logging events as strings of comma-separated entries into text files. Alternatively, interaction events can be logged to a database. Popular event logging mechanisms are widely known so I will not elaborate further on how to accomplish them. After implementing a logging mechanism and collecting some usage data, the analyst can then input the logged events into a visual interaction analysis system.

Although I outlined a guide and format to log events, this framework should also work with events logged in other formats. If a logged event format is different from what the framework requires, an analyst could convert the data into the required format for the analysis.

4.3 Visualization Design

Logged interaction events are next imported into a visual interaction system, IntiVisor [17]. IntiVsor includes five views that implement the visual interaction analysis framework. The layout of the system includes a control panel to the left and a main display area to the right that contains the visualization views. The data used in this section is from the CfD visualization. For more information about the CfD visualization, please reference Appendix A.

The control panel includes navigational buttons that support switching between all the visualization views included in IntiVisor, information about the data, and features for managing the analysis session (Figure 17). The navigational buttons, labeled Inspect, Select, Categorize, Sequence, and Explore, are in the upper part of the panel (Figure 17a). When the system starts, the bottom three buttons (Categorize, Sequence, Explore) are disabled. These buttons will progressively be enabled as an analyst provides the necessary information in their earlier views. I will present how these views work in the following sections.

Two pieces of information about the loaded interaction data are in the bottom part of the panel. First, the number of sessions (153) is next to the "Data Session" label (Figure 17b). Second, the session distribution is visualized in a bar chart (Figure 17c). The y-axis of the chart is time and the x-axis of the chart is the number of sessions. Each bar depicts the number of sessions in the corresponding time interval along the y-axis. The numbers next to the axes show the range of the data. An analyst may select a subset of the log data from a specific time range by dragging a time window with the mouse within this chart (gray box). This operation can be applied at any step of the analysis to filter data on demand.

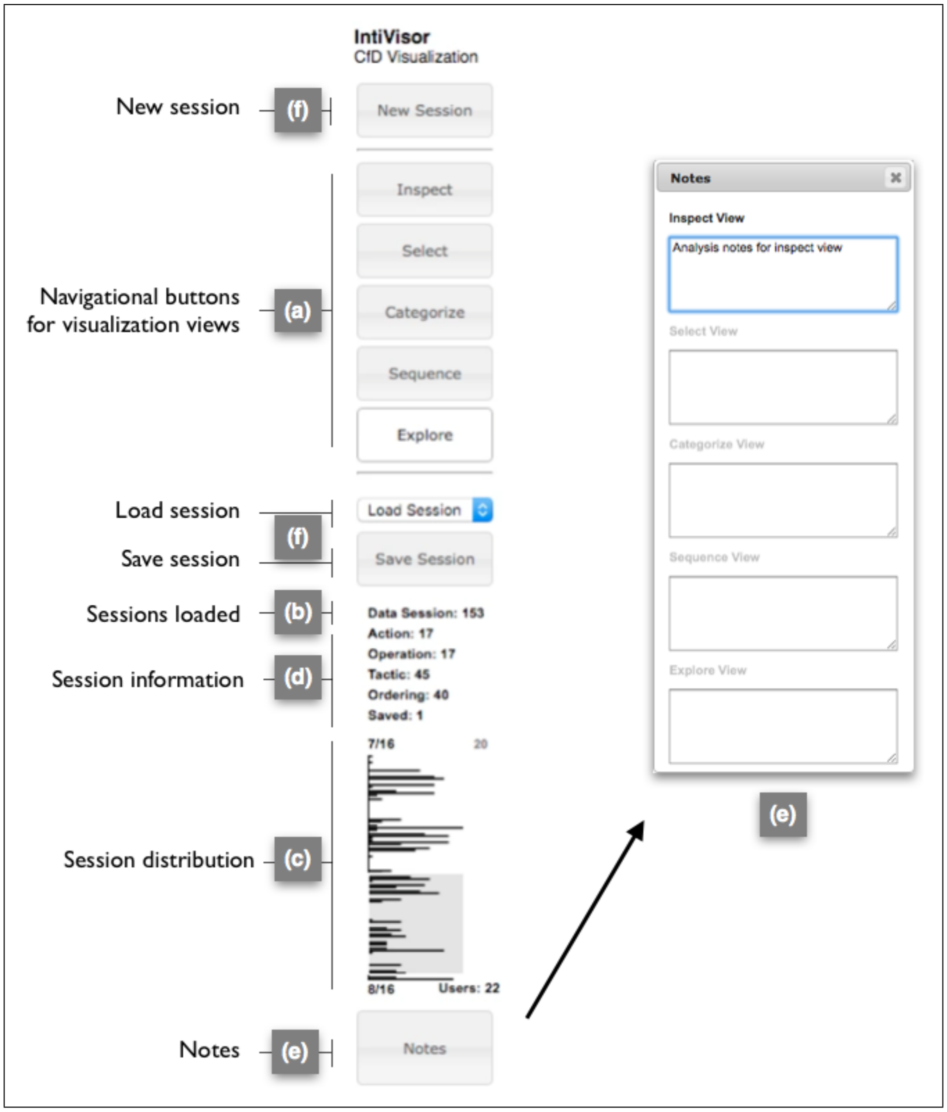


Figure 17: Control panel to the left of IntiVisor. (a) Navigational button for the visualization views. (b) Number of sessions loaded (153). (c) Session distribution over time. The highlighted area represents the time range of selected data. (d) Session information (e.g., Action, Operation). (e) Notes. (f) New, save, or load sessions.

Features for managing the analysis session are also available in the control panel. First, the number of analysis products from the framework, such as actions and operations, identified in IntiVisor are listed in the panel (Figure 17d). Second, IntiVisor supports note-taking in this panel. Clicking on the Notes button at the bottom brings out another panel that allows an analyst to take notes for each visualization view (Figure 17e). Third, the state of an interaction analysis session can be saved, loaded, or cleared (new) in this panel to allow an analyst to manage his/her analysis sessions (Figure 17f).

4.3.1 Inspect View

As with any log analysis, an analyst needs to gain a broad understanding of the logged interaction data to assess data quality and determine analysis directions. The Inspect View supports these tasks by allowing an analyst to inspect the logged interaction events sequentially over time with a scatterplot or a line chart. This view supports the Inspect part of the second step in the framework (Figure 16, step 2). The view includes two lists to the left, user and session, for an analyst to select which users' or which sessions' data to visualize (Figure 18). The left-most list is the user list. When one or more users in this list are selected, usage sessions from those users will be displayed in the session list to the right. Users are sorted by the number of sessions they have. This number, which is the total number of sessions linked to a particular user, is coded as a bar to the left of each user. The length of the bar shows the relative amount of sessions a user has. This visualization provides an analyst a quick view of the usage distribution by user.

The view includes three preconfigured visualizations: User Access By Time, Align Session by Time, and Align Session by Step.

1. User Access by Time

   This configuration shows a scatterplot of usage sessions over time by user, as shown in Figure 18. The x-axis is time and the y-axis is user. Each colored dot represents a usage session. By default, the color is redundant coding the user on the y-axis. It can also be mapped to other variables, such as session. Therefore, when using this view, an analyst can easily determine the usage distribution of the visualization application over both time and user.

2. Align Session by Time

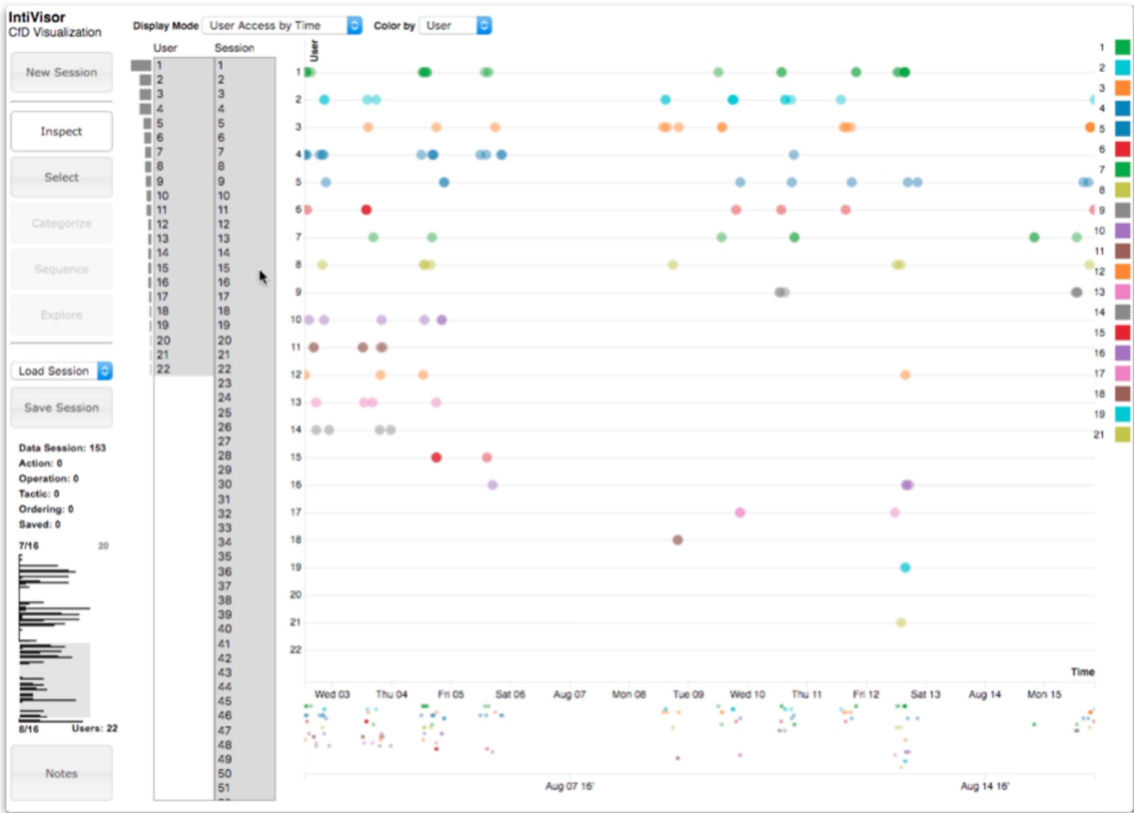


Figure 18: Inspect view: user access by time. The x-axis is time and the y-axis is user. This view shows that in about two weeks, about 2/3 of CfD visualization users regularly access the system on weekdays.

The x-axis can be reconfigured to show time relative to the beginning of each session, as shown in Figure 19. With this configuration, the first interaction of each usage session would be aligned to the left of the view. In this case, when multiple sessions are selected, their interaction patterns could be directly compared with each other from their overlapping patterns.

3. Align Session by Step

When the x-axis is time, some events may be clustered tightly or scattered loosely because actual time gaps between interaction events could vary significantly, as shown in Figure 19. As a result, this view makes it difficult to examine and compare patterns of event sequences. For a better look at the interaction patterns, I included a new x-axis unit called "step" that modified all time gaps

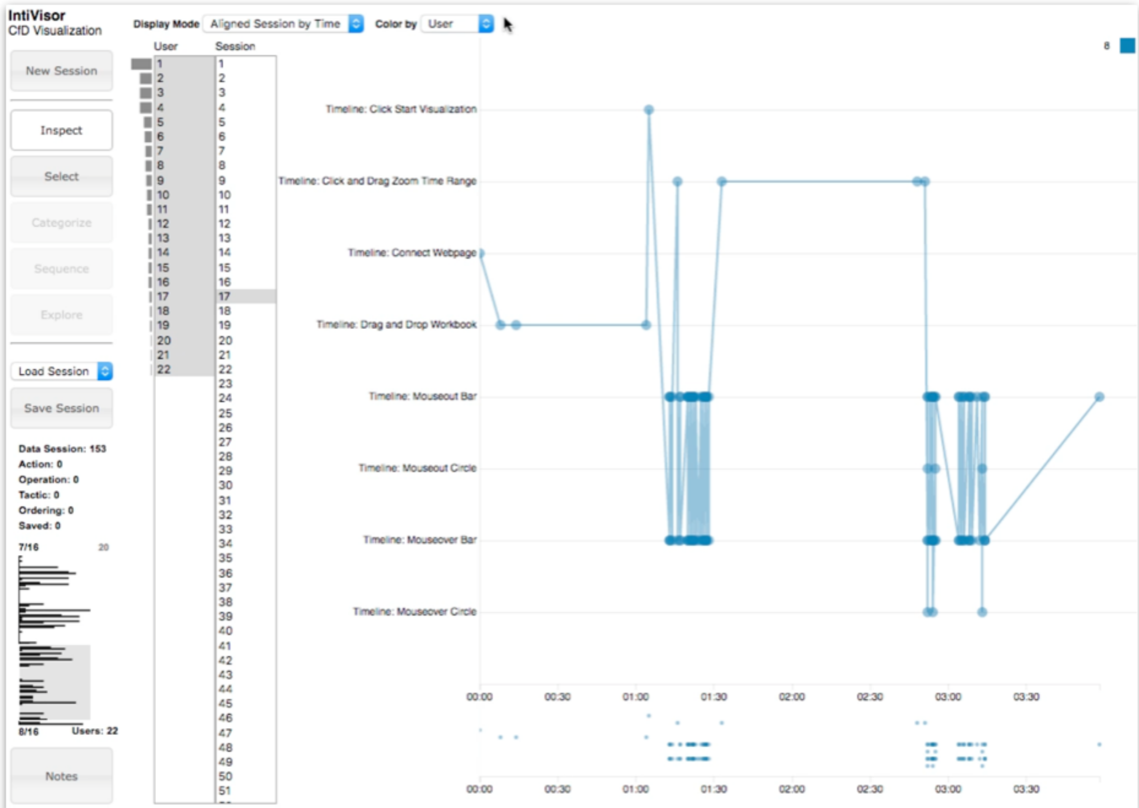


Figure 19: Inspect view: align session by time. The x-axis is time relative to the start time of the session and the y-axis lists all the events. This chart shows the interaction transitions over time.

into a fixed value, as shown in Figure 20. With this configuration, consecutive interaction events would be evenly spaced on the x-axis for a clearer view of interaction sequences.

4.3.2 Select View

Typically, an analyst would want to select, merge, or add parameters to events for their specific analysis goals. The Select view supports these tasks by implementing the Select part of the second step in the framework (Figure 16, step 2). Because not all events may be relevant to a given goal, an analyst may want to only select the ones that are relevant to minimize the amount of noise generated by irrelevant ones. Using this view, an analyst can select which individual events or event groups to include in

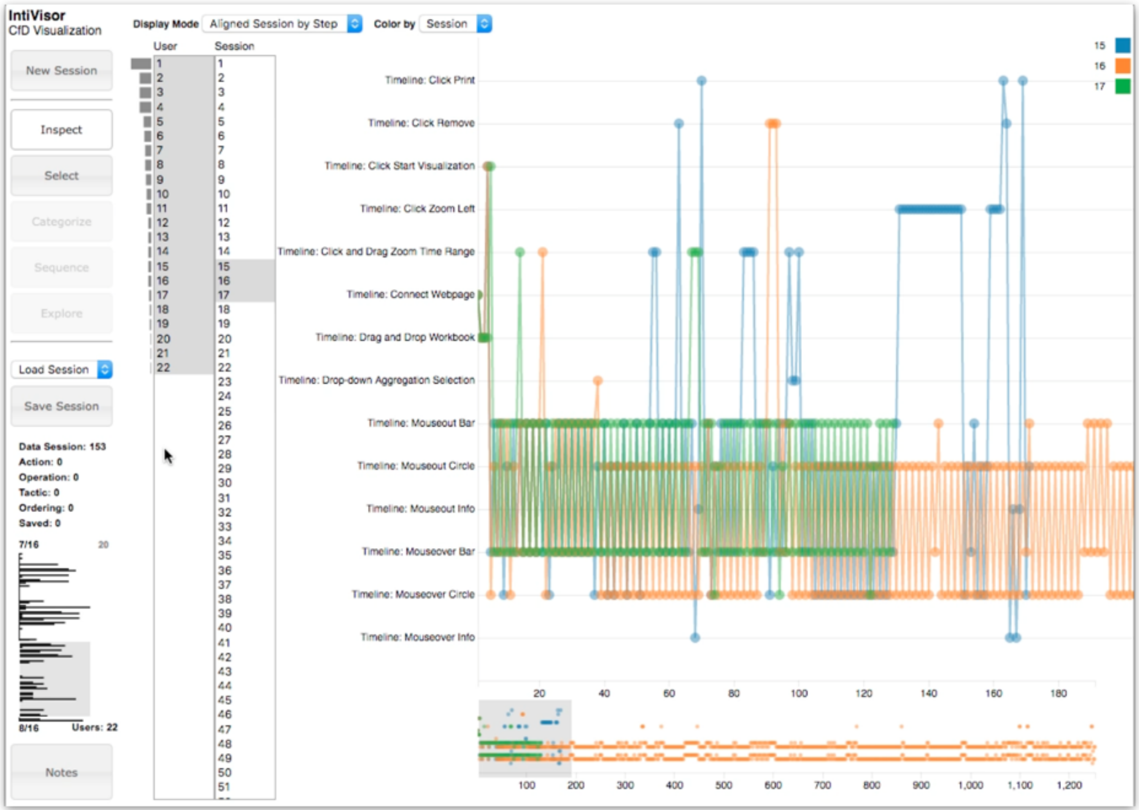


Figure 20: Inspect view: align session by step. The x-axis is "step," which is a unit that evenly spaces interaction events on the axis. The start of each session is aligned to the left of the chart. The y-axis lists all the events. This charts shows the interaction transitions in a clearer way because events are evenly spaced. Three sessions are shown in this figure.

the analysis. The left-most column is reserved for selected events (Figure 21a) and the three right-most columns are for unselected events or event groups (Figure 21bcd). Individual events are listed in the second column of the table and ordered by their occurrence frequencies (Figure 21b). The relative occurrence frequency of each event is shown as a bar and a percentage next to the event label. To select an event, an analyst clicks on the event in this column. IntiVisor will then move this event to the left-most column that keeps all the selected events with a yellow background (Figure 21a). Since many types of analyses would use a large portion of individual events, an analyst could click on the "Add All Events Below" button at the top of the second column as a shortcut to select all the events remaining in this list. If some batch selected individual events are not desired, the analyst could simply remove them after they are selected.

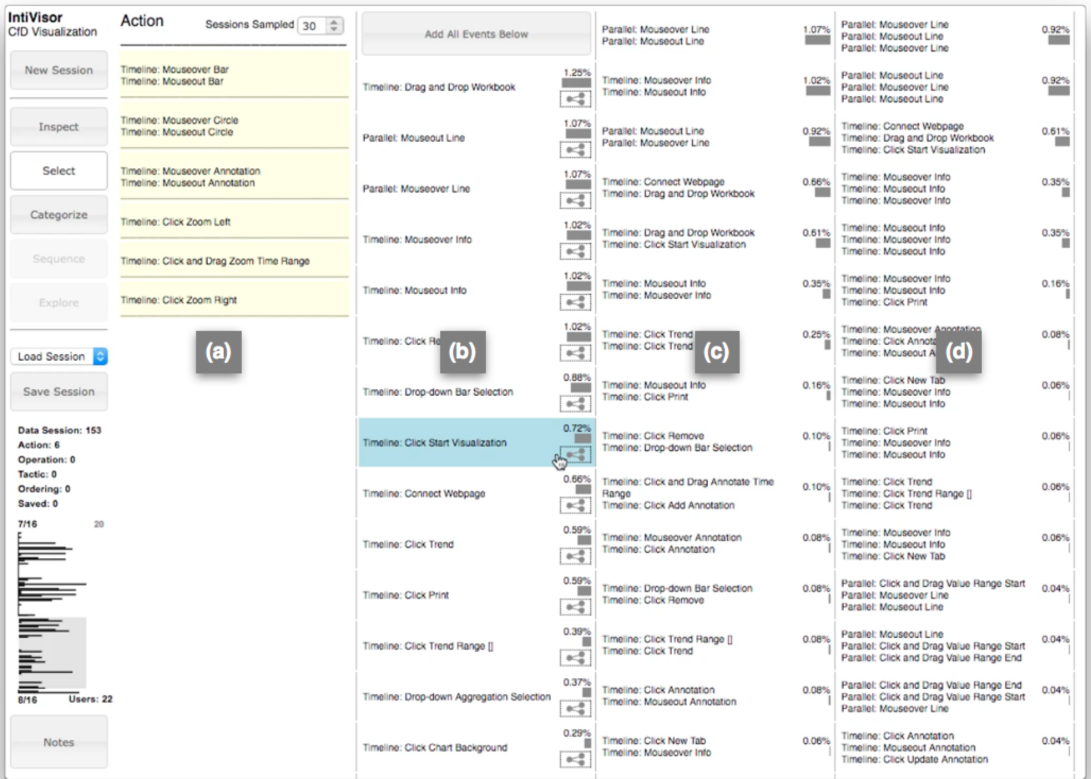


Figure 21: Select view. (a) Selected events and event groups. (b) Single events. (c) Pairs of events. (d) Triples of events.

Some events frequently occur together for accomplishing one single task, such as inspecting a visual data item with mouseover and mouseout events. In this case, an analyst may want to analyze these events together as a unit. Therefore, the Select View allows an analyst to select event groups of two or three consecutive events as a single unit, effectively merging them, for further analysis. Since meaningful event groups are typically used frequently, IntiVisor lists frequent 2- and 3-event event groups by their occurrence frequency in the two right-most columns, as shown in Figure 21cd. Similar as selecting events, an analyst can select event groups by clicking on them to move them to the left-most selection column. Because each selected event or event group likely represents a user action, they become “actions.”

When each action is selected, IntiVisor removes it from the entire dataset and re-extracts the frequent unselected event/event group lists. The reason why a re-extraction is necessary is because when an event/event group is removed, the frequencies of other events/event groups may also change. For example, if event A is removed, all frequent sequences that include A, such as AB, AC, ABC, will all disappear from the dataset. On the other hand, if a sequence ABC is removed, maybe only a subset of As, Bs, ABs, and BCs will disappear from the data because maybe not all of those events/event groups are in the sequence ABC. Therefore, as an analyst selects more actions, the unselected lists shorten. An analyst can stop selecting when seemingly all relevant events and event groups are selected. Because events and event groups are progressively removed from the unselected lists, the order of selection could influence the items remaining in the unselected lists. Therefore, if an analyst wishes to select an event group, it needs to be selected first. To ensure all event groups that an analyst wishes to select could be found, an analyst should start selecting the event groups with more events first, moving from right to left in this view.

Because the frequent event/event group lists are reconstructed after each selection, the response time becomes longer when the dataset is larger. To address this issue, I designed a sampling mechanism to only use a subset of the interaction data for generating the table in this view. I assumed that a reasonable sample would be able to provide an analyst enough information about the relative frequencies of events and event groups for selecting them. The relative frequencies are shown as percentages in the sampled data. For example, in Figure 21a, the event “Timeline: Click Start Visualization” occurred in 0.72% of events in the sample. The approximate number of sampled sessions can be manually configured and are shown at the top of the left-most column.

I sampled sessions using the following method. Different sessions have varying

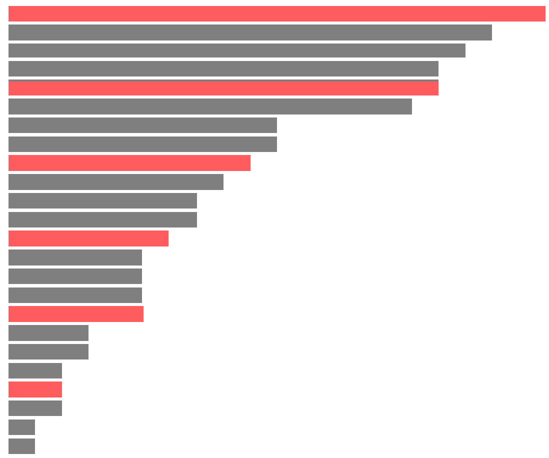


Figure 22: Sampling mechanisms in the Select view. Each bar represents a session. Parts colored red are the sample. When sampling for frequent sequences in the Select and Sequence views, the entirety of one out of n sessions are sampled up to the target sample number.

interaction amounts (lengths). I assumed that short sessions would have different interaction patterns than long sessions. As a result, to acquire a representative set of interaction events and event groups for an analyst to select, instead of randomly sampling the logged sessions, I systematically sample sessions by their lengths. This sampling process includes two steps. First, all the logged sessions are ordered by length (interaction amount, not actual time). Second, an approximate amount of sessions to the target sample size are selected from every one in n sessions, with n being the total number of logged sessions over the target sample size (Figure 22). For example, if the target sample size is 10 and the number of logged sessions is 99, then the algorithm will select every 1 out of 9 sessions (Math.floor(99/10)) from the sorted session list. As the list is sorted, the samples will have a similar length distribution as the original sessions. The number of sampled sessions will be 99/9 = 11, approximately the target sample size. Note that for simplicity, I did not explicitly make sure the sample size is exactly the same as the target sample size. In this step, an approximate number is sufficient because only a percentage of interaction occurrence will be displayed in the view.

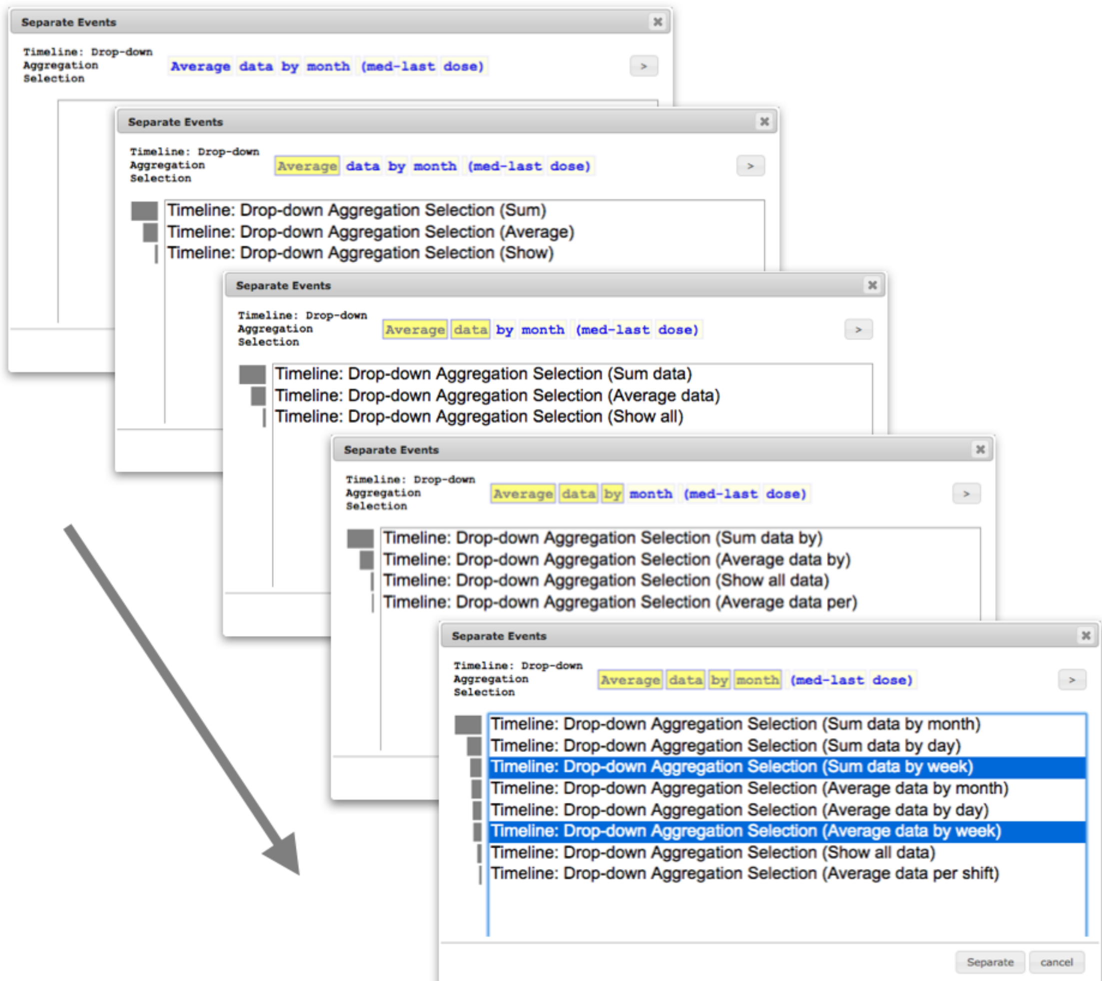


Figure 23: Separate Events dialog for adding parameters to events. As an analyst selects parameters (0→4, top→bottom, highlighted yellow), parameter values for the selected parameters will be listed in the bottom panel. Their relative occurrence frequencies are displayed as bars to the left. In the last view in the bottom, two events attached with the specific parameter values, "Sum data by week" and "Average data by week" are selected for extraction.

The default Select View only displays interaction events without their parameters. However, many events include parameters that may be necessary to include for analysis. For example, when clicking a toggle button, it turns a switch on and off. The on and off state is a parameter of the interaction event that could be of interest to an analyst. To include parameters to the analysis, an analyst selects the button in the event entry to which he/she wishes to add parameters. The "Separate

Events" dialog will appear that shows two panels (Figure 23). One panel at the top displays the event with its parameters from one of its log entries. An event can have any number of parameters. Typically, multiple parameters are delimited by several commonly-used delimiters, such as space or comma. IntiVisor does not detect which delimiter is used. It simply breaks the parameter list by all the common delimiters to support the extraction of individual parameters. For example, if a "zoom in" event has parameters that define the focus point (x,y) of the zoom and the zoom level, these parameters may be delimited by a comma and a space such as "10,10 2." When the two delimiters are applied, the three parameters—"10", "10", and "2"—are extracted. In the top panel of the Separate Events dialog, the automatically extracted parameters are highlighted in yellow boxes.

To select parameters to include in the analysis, an analyst selects the corresponding yellow boxes. For example, if only the zoom level parameter is of interest to an analyst and not the zoom focus coordinates, the analyst simply selects the last parameter. But what if the parameter delimiter incorrectly segmented the parameter list? For example, in Figure 23, one parameter was incorrectly segmented into multiple ones because the white space was not used as a delimiter in the event. In this case, an analyst can select multiples of these incorrectly segmented parameters to "combine" them into the correct one, as shown in the figure. When a parameter is selected, all events of this type having this parameter will be listed in the bottom panel. The events are listed in descending order by the frequency of the parameter value. The relative frequencies are shown as a bar chart to the left of the list. When more than one parameter is selected, the list will be updated and sorted accordingly based on the combination of the parameter values. An analyst can next select the events with specific parameter values to extract. Multiple events can be selected at the same time. For example, if an analyst only cares about zoom level 2 and 0.5, just these parameter values are selected. When the selection is complete, events that include those selected parameter values will be extracted from the original event as new event entries in the single event list. If an analyst only selected a subset of events with specific parameter values, such as in the zoom level example, the original event will still be left in the single event list because events with other zoom levels are still not separated from it. A star will be attached to an event (e.g., zoom in*) that had some events (with specific parameters) separated from it to differentiate it from those without separated events in the unselected event list. If later the analyst decides that a separated event with specific parameters is no longer required to be separated from the original event, the separated events can be merged back into the original event.

Sometimes, logging formats are updated. For example, an event that formerly had have two parameters subsequently may have three. Therefore, if the default event shown in the top panel only includes two parameters, there is no way to select the third parameter for events that were logged after the update. In this case, an analyst can click the ">" button on the right of the parameter list to select a different event sample that includes the third parameter.

4.3.3 Categorize View

After actions are selected, an analyst can group them into a smaller, more semantically meaningful set of categories to further organize and interpret the interaction data (e.g., [13, 16]). But how should actions be categorized? In IntiVisor, an analyst needs to create an *analysis perspective* for this in the Categorize View. This view implements the third step of the framework (Figure 16, step 3). An analysis perspective defines a set of categories or "operations" in the framework based on, for example, tool feature or user intent. But how should a perspective be defined in IntiVisor? IntiVisor provides a canvas that allows an analyst to layout his/her categories to create a perspective. Using a spatial layout is how people naturally organize things for memory and semantics [2]. To further magnify the benefits, IntiVisor supports placing items around labels or certain objects to help remind about the categories of these items. To utilize this organization method, IntiVisor provides a drawable canvas in the Categorize View to help an analyst create a context for organizing categories in a perspective. This view includes drawing tools that can support drawing lines, rectangles, circles, and texts onto the canvas (Figure 24). The drawings provide a context to the perspective and visually define a set of spatial constraints for the categories. For example, if an analyst chooses to define a perspective with a category for each tool feature on the UI, a set of drawing tools can be used to sketch the UI layout as the context of the analysis perspective because each category would correspond to one feature that maps to a drawn UI layout position (Figure 25a). The UI layout does not need to be accurately drawn to scale. It only needs to provide a recognizable context to the analyst for categorizing events. Alternatively, an analyst could choose to import a screenshot of a visualization application as a background image by dragging and dropping a screenshot image into the rectangular box (light gray to the left of "Hold Shift to Remove") labeled "Image" in the view (Figure 25b).

An analysis perspective can be determined by a predefined set of categories. For example, the interaction categories defined by Yi et al. provides a set of categories that correspond to user intents (e.g., select, filter). To define this perspective, an analyst can draw several regions, each representing one of the interaction categories, as shown in Figure 25c. In the figure, I further divided the regions, which are the ones with labels, by where they are likely to occur in a typical chart layout. For instance, the Encode and Filter categories are listed at the left of the chart area because they typically occur in control panels to the left of a chart. Similarly, the Select and Abstract/Elaborate categories are listed within the chart area because that is where those interactions usually occur. A predefined set of categories can come from known analysis methods. For example, the Visual Information-Seeking Mantra includes four user activities: overview, zoom, filter, show details [41]. These

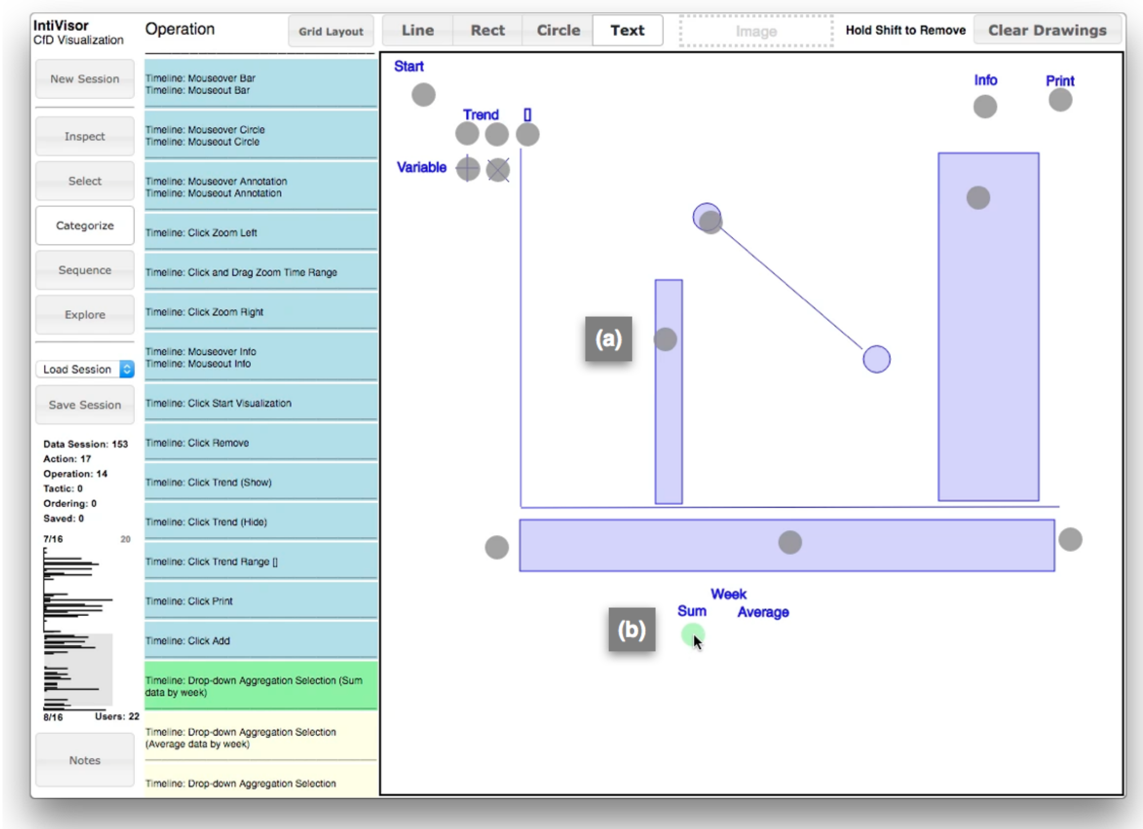


Figure 24: Categorize View. (a) Action assigned to bar area. (b) Action assigned by dragging and dropping a circle onto the canvas.

activities can be the categories if an analyst seeks to identify the use of the mantra. The mantra represents a top-down analysis method that starts from inspecting all the data with less individual details (high-level) to inspecting details of individual data items (low-level). As a result, in Figure 25d, I drew the Overview category at the upper part of the view and the Details category at the lower part of the view. Zoom and Filter were kept in the middle as they could occur in any order.

After an analysis perspective is defined, an analyst next decides which actions map to which categories. In the Categorize View, all the selected actions are listed at the left side of the view. The analyst needs to drag and drop each action onto the canvas to assign it to a category as an "operation." For example, to assign the action "Timeline: Drop-down Aggregation Selection (Sum data by week)" onto the canvas

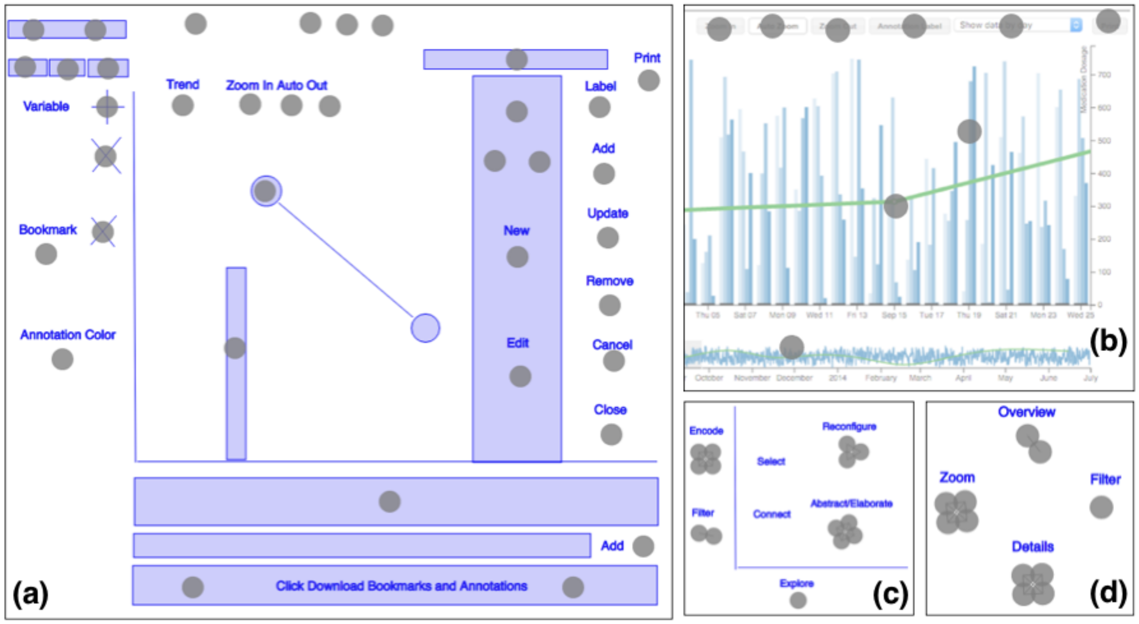


Figure 25: Analysis perspectives. (a) Categories from features mapped to drawn UI layout. (b) Categories from features mapped to UI image. (c) Categories from interaction categories by Yi et al.'s [50]. (d) Categories from the Visual Information-seeking Mantra [41].

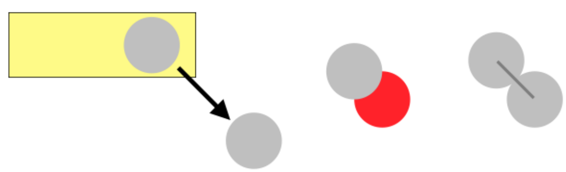


Figure 26: Categorizing multiple actions into an operation. An action is dragged from the list at the left side onto an operation (assigned action) on the canvas to categorize them into the same operation.

as an operation, the analyst should drag and drop the action, which is represented by a gray circle, next to the label "Sum" in Figure 24b. In the list to the left, unassigned actions have a yellow background whereas assigned actions have a blue background. Similarly if an analyst wants to assign an interaction with a visualized bar, he/she could drag the corresponding action onto a visual "bar area" for this categorization (Figure 24a). The spatial positions of the assigned actions (circles) only need to be accurate enough for the category assignment to be clear to the analyst. If multiple actions are to be assigned to the same operation, an analyst can drag and drop an action onto a suitable location on the canvas first and then drag and drop the other actions from the list onto the assigned actions, as illustrated in Figure 26. This assignment will create links between the assigned actions (circles) that indicate, going forward, these actions will be seen as the same new operation in IntiVisor. Most operations in Figure 25cd have more than one action categorized into them.

Since the assignment of actions can be a labor-intensive step, a shortcut is available for automatically placing unassigned actions into a grid layout on the canvas, irrespective of existing action assignments and the perspective. This shortcut, available with the "Grid Layout" button above the list of actions, allows an analyst to quickly assign actions. The downside is that the location of the actions are not as meaningful to an analyst anymore because they are automatically assigned. Therefore, a mixed use of the assignment methods could provide a nice balance of meaningful, contextual action placements and reduced labor. For example, an analyst can start assigning actions they care more about into contextually meaningful locations on the canvas. Afterwards, the remaining actions can be assigned automatically using the grid layout.

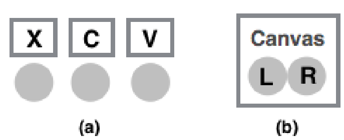


Figure 27: Solutions to challenges in assigning actions to a UI layout. (a) Actions that do not have a spatial location to map to, such as keyboard shortcuts, could be assigned to a dedicated region on the canvas. (b) Actions that occur in the same spatial location, such as left- and right-clicking mouse buttons, could be assigned side by side.

As mentioned in the related work section, there are challenges in mapping actions to their corresponding spatial locations in a visualization application's original UI layout. First, where should an interaction that does not have a clear corresponding spatial location on the UI layout, such as pressing a keyboard shortcut, be positioned? A simple solution is to isolate an area on the canvas that is dedicated to keyboard

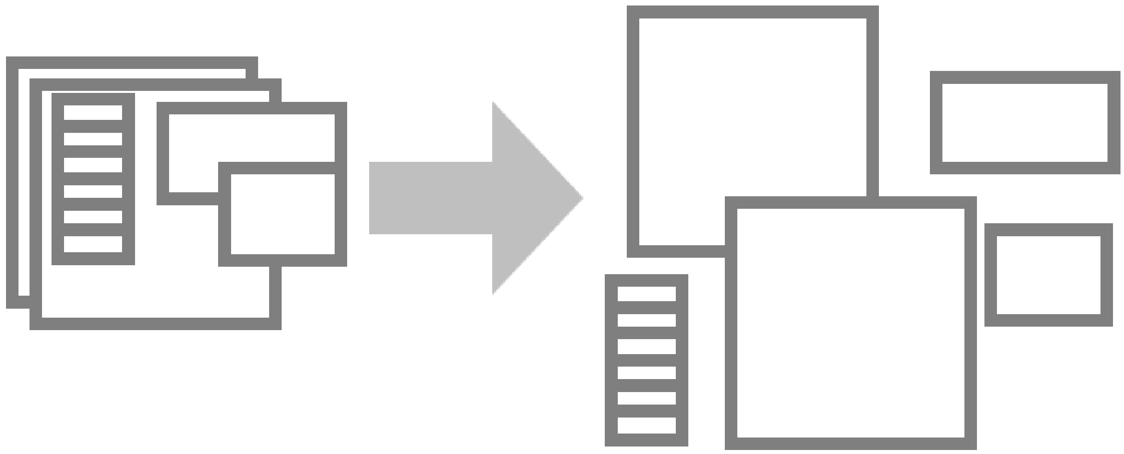


Figure 28: Unpacking a layered set of windows, dialogs, and menus onto a 2-D layout.

shortcuts, such as in Figure 27a. Or an analyst could choose to place the keyboard shortcut next to a button that provides the same feature. Second, the same spatial location may accept different types of user interactions, such as left- and right-clicking mouse buttons. In this case, an analyst can choose to map these actions side by side next to the corresponding UI position in a consistent manner across the perspective. For example, if left- and right-clicking a canvas area have different effects, an analyst can draw a rectangle representing the canvas area and move the left- and right-mouse-clicking actions on top of the rectangle area but next to each other, as shown in Figure 27b. If an analyst prefers, he/she could label the actions and canvas background for a clearer presentation. Third, what if a visualization application has multiple views, dialogs, and menu items? This challenge is fundamentally about mapping a 3-D dataset (layered views) onto a 2-D space. A simple solution is to unpack the layers, as shown in Figure 28. The original layered UI views are unpacked into five mostly side-by-side views. All the actions on those views can now be mapped to a 2-D UI layout. Note that for more complex visualization applications that have over dozens of such views, available space on the canvas can run out before all the UI views are drawn. In this case, an analyst could selectively draw a subset of the views that carry interactions that are of most interest, or merge actions on different views when they do not need to be separated in the analysis.

IntiVisor currently supports assigning a maximum of 40 operations because the system uses colors to differentiate operations in a later view. The operations would not be as visually differentiable if there are too many of them. This limitation may not be very significant because most studies used less than a dozen of categories (e.g., interaction categories by Yi et al. [50]) to organize events. Therefore, 40 operations should be sufficient for most analyses. If an analyst attempts to assign more than 40 operations, an error message will appear to inform him/her that the limitation is reached. Two solutions are available when an analyst runs into this limit: (1) an analyst can consider merging some actions by categorizing them into the same operation, or (2) an analyst can consider disregarding a set of operations that are of less interest to the current round of analysis. After categorizing actions into operations, the next step is to start the analysis by computationally extracting frequent sequences of operations.

4.3.4 Sequence View

Because individual operations are not sufficient to help identify higher-level usage patterns, an analyst next needs to find sequential patterns from them. Frequent sequences of *consecutive* operations, which I call "tactics," are automatically extracted from the data in the Sequence View. This view implements the fourth step of the framework (Figure 16, step 4). Sequences are shown as triangular, tapered edges between circles, which represent operations, in a graph view (Figure 29). To illustrate the design, see Figure 30. Operations A, B, C form a sequence and operations B, C form another sequence. An operation with a larger number of actions mapped to it are drawn larger in size, such as operation C. Because different sequences, such as ABC and BC in the figure, may have edges between the same pair of operations (BC), the pointy end points of the edges are randomly jittered so that the edges won't always exactly overlap, allowing multiple overlapping edges to be visible at the same time. Moving the mouse cursor over either a node or an edge will highlight all sequences that include the node or edge. This feature allows an analyst to quickly discover how frequently used a given node or edge is in the dataset. All the extracted sequences are listed at the left side of the view. An analyst can move the mouse cursor over the list items to highlight the corresponding sequences. He/she may also select one sequence by clicking on it in the list and then navigate through the list with up and down keys.

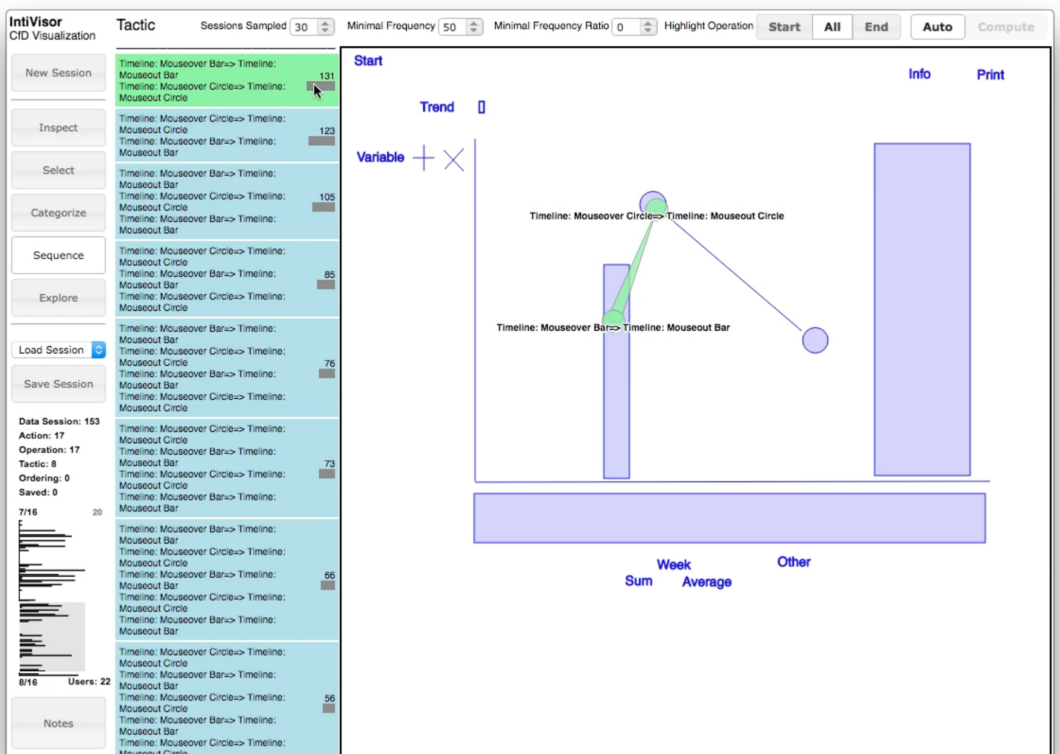


Figure 29: Sequence view showing frequent sequences of consecutive operations. One sequence is highlighted.

The frequent sequences of consecutive operations to be extracted can be configured in this view. Several options are available:

- Sessions Sampled

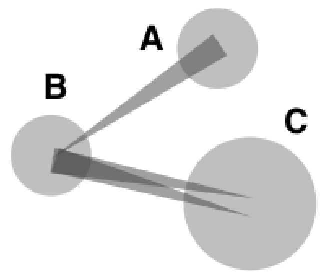


Figure 30: Sequence representation. A, B, and C are operations. C is bigger because it is mapped to more actions.

For the same reasons discussed in the Select view, by default IntiVisor only extracts frequent sequences from a sample to avoid lengthly computation time when the dataset is large. The sample size is set the same way as in the Select view. An analyst can manually configure this number in the upper-left corner of the view.

- Minimal Frequency

  An analyst typically seeks to identify widely used visual analysis patterns from frequent sequences. But how "frequent" should a sequence occur to be considered "frequent?" An analyst can set a *minimal frequency threshold* for this purpose. Sequences that occur less frequent than the threshold are filtered out. This parameter is directly related to the sample size because the more session samples there are, the higher this value needs to be set to filter out the same portion of sequences.

- Minimal Frequency Ratio

  Although frequent sequences can identify typical analysis patterns, sometimes the most unexpected, and thus insightful, patterns are not the ones that occurred the most frequent. For example, some operations occur significantly more in the dataset. When an operation is pervasive, chances are that sequences including it are more likely to be frequent. Therefore, these sequences being frequent may not be surprising to an analyst. For example, in Figure 31a, a set of "frequency sequences" are displayed in a graph. The numbers in each circle (operation) and edge (sequence) represent their corresponding occurrence frequencies. For example, in this simple illustration, every sequence occurred 5 times so they are all about the same width (length is irrelevant). One circle is bigger than the others, indicating that the operation it represents occurred more (20 vs. 5). Note that this illustration is not mapping the circle sizes in the same way as in the Sequence view. It is only used for illustrating this filter. The key takeaway from Figure 31a is that many frequent sequences include the prevalent operation (large circle). This phenomenon is intuitive as the chance for a frequent operation to be in a frequent sequence is higher. But this type of frequent sequences may be less interesting to an analyst. They may occur so often that they obscure other frequent sequences that might have been more interesting.

A "frequency ratio" is a metric for separating these types of sequences. As illustrated in Figure 31b, it is a ratio calculated by dividing the frequency of a sequence by some combination of the frequencies of its composing operations. For sequences with consecutive operations, they are calculated with the equation $\frac{x}{\sum y_i}$, where x is the occurrence frequency of a sequence, y is the occurrence frequency of an operation in the sequence, and i is the ith operation in the sequence. With this metric, sequences with lower frequency ratios would be the ones that have relatively higher sequence occurrence frequencies than the frequencies of their composing operations. For example, Figure 31c shows a list of sequences that occurred the same amount but with different combinations of operation frequencies. The sequences' relative frequency ratios are listed in Figure 31d. From Figures 31cd, it is clear that frequency ratios are higher when a sequence includes operations with higher frequencies. When a minimal frequency ratio is set as a filter, it can filter out sequences that include the large circle (more frequent operation), leaving the remaining, potentially more interesting frequent sequences to stand out.

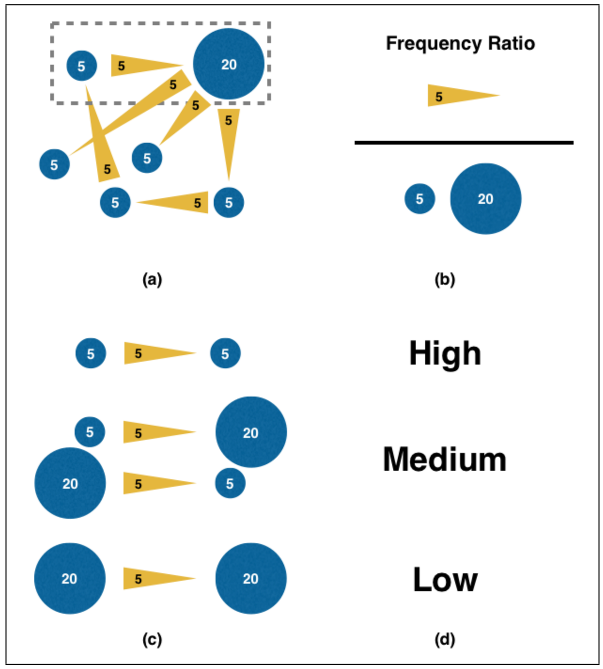


Figure 31: Minimal frequency ratio is a filtering threshold for removing sequences that include operations that are prevalent in the data. The numbers and sizes of the operations and sequences are proportional to their occurrence frequencies. (a) More frequent sequences include the operation that occur more. (b) Minimal frequency ratio is defined by the frequency of a sequence over a combination of frequencies of the composing operations. (c) A list of sequences that have different combinations of more/less frequent operations. (d) The relative minimal frequency ratio for the sequences. As a result, when the threshold of a minimal frequency ratio is set higher, sequences that include operations with relatively higher occurrence frequencies will be filtered out.

- Highlight Operation (Start/All/End)

  An analyst may be more interested in sequences that include a specific operation. For example, an analyst may want to explore what users frequently do after zooming. To accomplish this task, an analyst can use a specific operation to filter frequent sequences in the Sequence view by clicking on that operation, "Timeline: Click and Drag Zoom Time Range," as shown in Figure 32. The operation will be highlighted in yellow and by default, sequences that do not include it will be filtered out. This behavior is used when the default setting for highlighting operations is set to "All," which means all sequences that include the operation is kept for the analysis. Alternatively, an analyst can further specify the selected operation to be at the "Start" or "End" of a kept frequent sequence. So in the example, an analyst would specify that "zoom" should be at the "start" of every sequence to find which activities frequently occur after it.

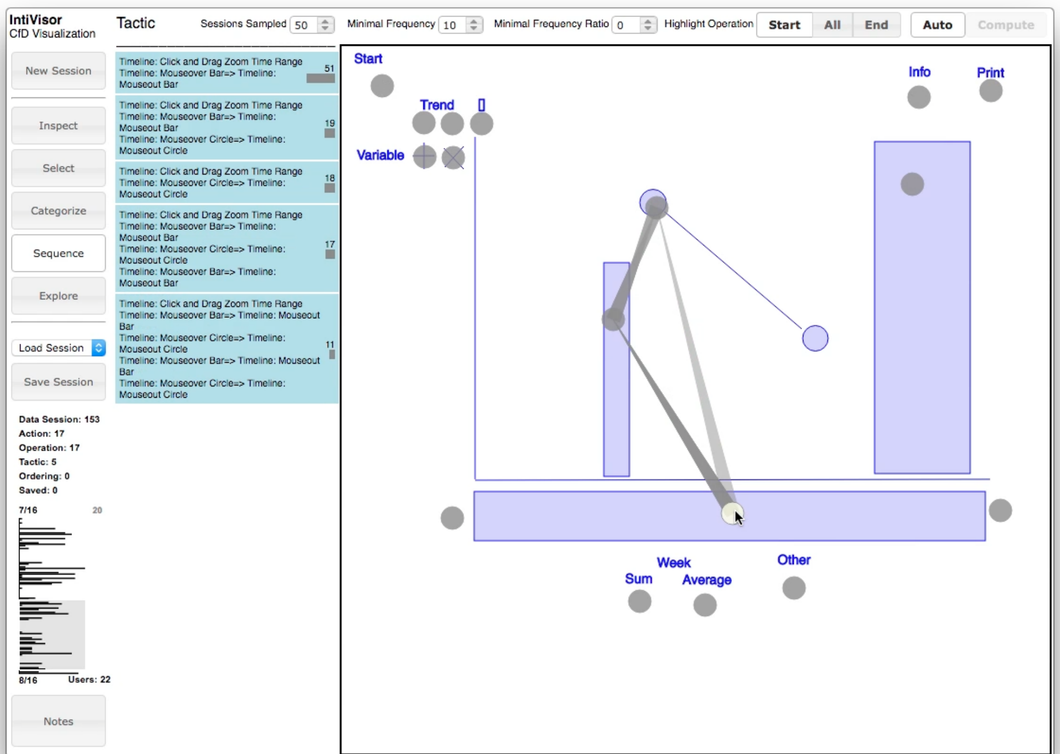


Figure 32: Sequence view showing only frequent sequences that starts with the operation "Timeline: Click and Drag Zoom Time Range."

- Auto Compute

  Related to performance, by default every configuration change automatically causes a re-extraction of sequences. However, when the sequence extraction is too slow, an analyst may not want the system to automatically extract sequences upon every configuration change anymore as it interrupts the analysis process. By unchecking the button "Auto," an analyst can disable the auto compute feature and manually use the "Compute" button next to it to extract sequences on demand.

A major assumption in this view is that sequences of the same operation, such as AA or AAA, are less interesting to an analyst. Interesting patterns should include a combination of interactions, such as AB or ABA. As a result, I filtered out all tactics that include consecutive operations that are the same. Once a meaningful set of tactics are identified, an analyst next moves to the last view that puts all the event organization outputs together into one view.

4.3.5 Explore View

Once events are organized into operations and tactics, an analyst would need a view to inspect them together to find even higher-level usage patterns and analysis methods (Figure 16, step 5). The Explore view supports this task by not only integrating operations and tactics into multiple connected visualization views but also supports the identification of frequent sequences of non-consecutive operations, which I call frequent orderings. The view includes several visualization subviews: (1) Graph view, (2) Bar view, (3) Usage distribution view, and (4) Frequent orderings view. The subviews are connected where interactions in one view can cause simultaneous updates in other subviews.

4.3.5.1 Graph View

The graph view provides an overview of all the operations and tactics identified from previous steps in the analysis process in their analyst-defined context. The view is located in the upper-left corner of the Explore view (Figure 33a). Visually, it is a smaller version of the Sequence view but each operation has a unique visual

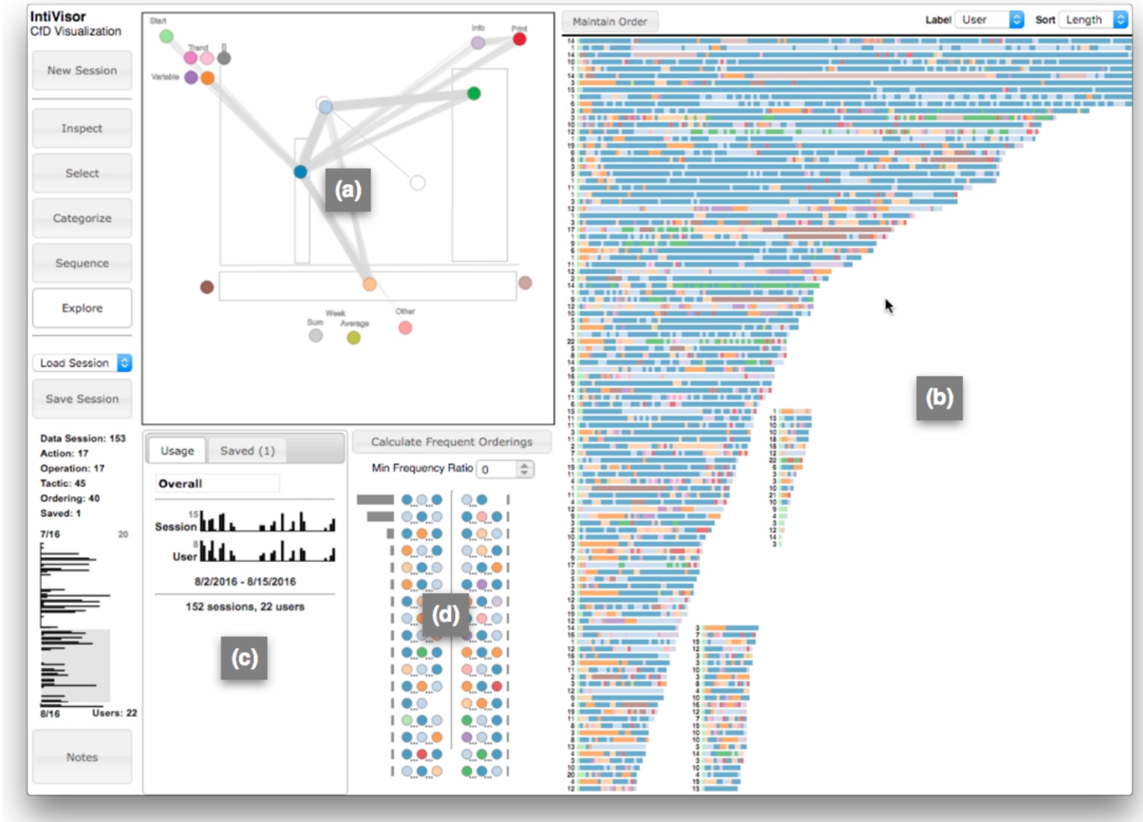


Figure 33: Explore view. (a) Graph view. (b) Bar view. (c) Usage distribution view. (d) Frequent orderings view.

representation. The first 20 operations assigned are colored by 20 differentiable colors. The following 20 operations are colored by the same 20 colors but with a noticeably thick border (Figure 35a). The unique visual representations allow each one of the operations to be easily identifiable when also shown in other views. Because the circles in this view were laid out by an analyst, their mappings to operations should be instantly recognizable to the analyst by its spatial location and background context. Therefore, the view also becomes a “legend” for an analyst to identify operations in other views that has the same (frequent orderings view) or similar (bar view) visual representations. These other views will be presented in the next sections.

An analyst can select operations and tactics in the graph view to display in the other views. To select an operation, an analyst clicks on the circle representing the

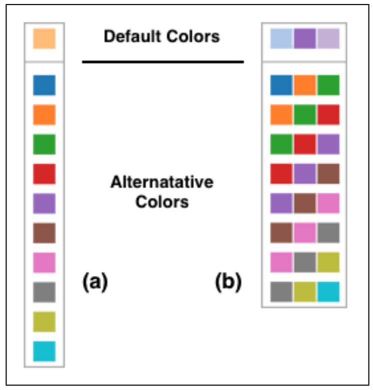


Figure 34: Color selection menu available in the graph and bar views. The default color selected by the system is shown at the top. Other selectable alternative colors are listed below the default color. For individual operations, the menu supports the selection of one color (a). For tactics that includes multiple operations, the menu supports the selection of multiple colors in groups (b).

operation. A color choice menu will appear that allows an analyst to optionally specify a different highlight color other than the default color (Figure 34a). The reason why the system allows an analyst to optionally select a different color is because some of the lighter default colors do not stand out as much for visually highlighting items. As a result, IntiVisor provides a set of highly saturated colors as options for highlighting an operation in addition to the default color (Figure 34a). When a highlight color is set, the selected operation will be colored with it. Alternatively, to select a tactic, an analyst clicks on one of the edges of the tactic. A color choice menu with groups of colors will appear that similarly allows an analyst to optionally select a different set of colors other than the default colors for highlighting the set of operations in the tactic (Figure 34b). When a set of colors are selected for the tactic, the operations in the tactic will be colored accordingly. IntiVisor can highlight the pattern selected, both in this view (graph) and the bar view to the right. Because the selections are closely tied with the bar view, I will present the details on how they work in the next section when I introduce the bar view.

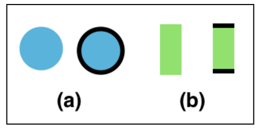


Figure 35: Visual encoding of operations in the graph view (a) and in the bar view (b) of the Explore view.

4.3.5.2 Bar View

The bar view provides an analyst a visual overview of all the operations organized by sessions. This view is located at the right side of the Explore view (Figure 33b). It presents all the operations by session in a set of horizontal bars. Each bar represents a session and is composed of a set of blocks, each representing an operation in the session. Every operation is colored with the same color used in the graph view. Similar to the graph view, the second 20 operations have a pair of thick lines but are above and below the blocks to differentiate them from the first 20 operations (Figure 35b).

By default, sessions are sorted vertically by their horizontal length (number of operations). I designed a special layout for this sorting method to maximize the utilization of screen space. As shown in Figure 36 when the list of sessions reaches the bottom of the display, the remaining sessions (red bars) that should have been below the visible area, are moved up to the available white spaces to the right. This process can be iteratively applied to each new set of bars that would have been below the visible area. Using this layout mechanism, when the amount of data increases, the visualization expands horizontally instead of vertically. Figure 37a shows how this layout appears in the bar view. With the prevalence of widescreen displays, this layout mechanism increases the likelihood that more sessions and operations can be shown on screen. Sessions can be sorted by other criteria, such as time and user. When sessions are sorted by time, they are listed chronologically by their start time (Figure 37b). When sessions are sorted by user, they are grouped and listed alphabetically by user (Figure 37c). When sessions are sorted by a criterion other than length, the layout mechanism for sorting by length will no longer be applicable.

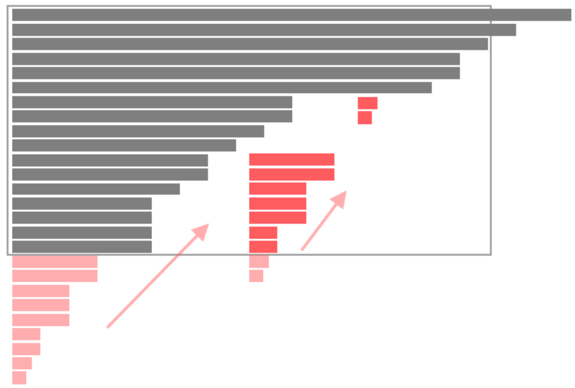


Figure 36: Layout mechanism for sorting sessions by length. Sessions (bar) listed below a certain level (e.g., display area, rectangle frame) is moved to the blank space to the right.

Figure 37: Sorting Methods. (a) Sessions sorted by length. (b) Sessions sorted by time. (c) Sessions sorted by user.

By default, sessions are labeled with their users on the left side of their bars (Figure 38a). Moving the mouse cursor over a label will temporarily hide other labels that are not of the same value. If the labels are users, this interaction provides an analyst a quick view of which sessions belong to a highlighted user. Note that only

(a) (b) (c)

Figure 38: Labeling sessions. (a) Labeled by users and sorted by length. (b) Clicking on one user label temporarily hides sessions of other users. (c) Both labeled and sorted by user.

the labels are temporarily hidden with this interaction, not the colored bars. If an analyst wishes to filter out the colored bars that do not belong to the highlighted user, a click on the label will help achieve this goal (Figure 38b). The resulting view only displays colored bars that belong to the highlighted user. If the sessions are also sorted by users, the view can help show user-specific interaction patterns (Figure 38c). The labels can be changed to other available metadata about the sessions, such as the usage "reason" in the CfD visualization, if an analyst seeks to look for relations between usage reasons and the data.

Several configurations are available from the context menu of this view:

- Show/Hide Unselected Events

  Because only events that are selected and categorized are displayed in the bar view, two operations that seemingly occur next to each other in this view does not necessarily mean that they occurred consecutively in the original usage session. Therefore, to examine how many logged interactions are not included in the view, an analyst can use the "Show/Hide Unselected Events" feature in the context menu of the bar view. If some events were not selected, where they would have been in the bar view would be left with a white space (Figure 39a → Figure 39b). This way an analyst can gauge how much activity was there between every given pair of consecutive selected operations.

(a) (b)

Figure 39: Show unselected events. (a) Default view. (b) Show unselected events.

- Width Size

  The width of each operation in the bar view is adjustable. When sessions are short, some white space may be available to the right. In this case, an analyst can increase the width of operations to extend the horizontal size of the visualization to better utilize space (Figure 40a → Figure 40b). This increase in size can make it easier to inspect and select individual operations. On the other hand, when sessions are long and extend beyond the screen width, an analyst can reduce the size of operations to fit more of them in the visible area.

- Merge/Separate Consecutive Operations

(a)

(b)

Figure 40: Increase width size. (a) Default view. (b) Increase width size.

An operation may be used repeatedly in succession. If an analyst does not care about repeated operations, he/she can merge consecutive operations of this type in the bar view (Figure 41a → Figure 41b). For example, a user may repeatedly click a button to adjust the zoom level of a view. If an analyst does not care about how many times the zoom level was adjusted, these consecutive zoom level adjustment operations could be merged into one zoom operation for the analysis.

An analyst can select operations from the bar view the same way he/she selects it from the graph view. A color menu will appear in the selection area when the analyst clicks on any of the colored blocks. When an operation is selected, a colored circle that represents it will appear above the bar view (Figure 42a). Selections from the graph view also shows up in this area. Recall that in the graph view, tactics are also selectable. When a tactic is selected, partially overlapping circles that

(a) (b)

Figure 41: Merge consecutive operations. (a) Default view. (b) Merge consecutive operations.

indicate consecutive operations, are shown in this area. Multiple types of selections are supported in IntiVisor as follows:

- Independent Selection

  The simplest form of selection is selecting individual operations and/or tactics independently. An analyst can use this selection method to highlight all occurrences of these operations and tactics to inspect their overall distribution in the log data within and between sessions. IntiVisor supports selecting individual operations and tactics from the graph view or operations from the bar view. By default, when a set of operations and tactics are selected, they are highlighted in the bar view independently. Multiple selections are separated by white spaces in the selection area, as shown in Figure 43a.

- Ordered Selection

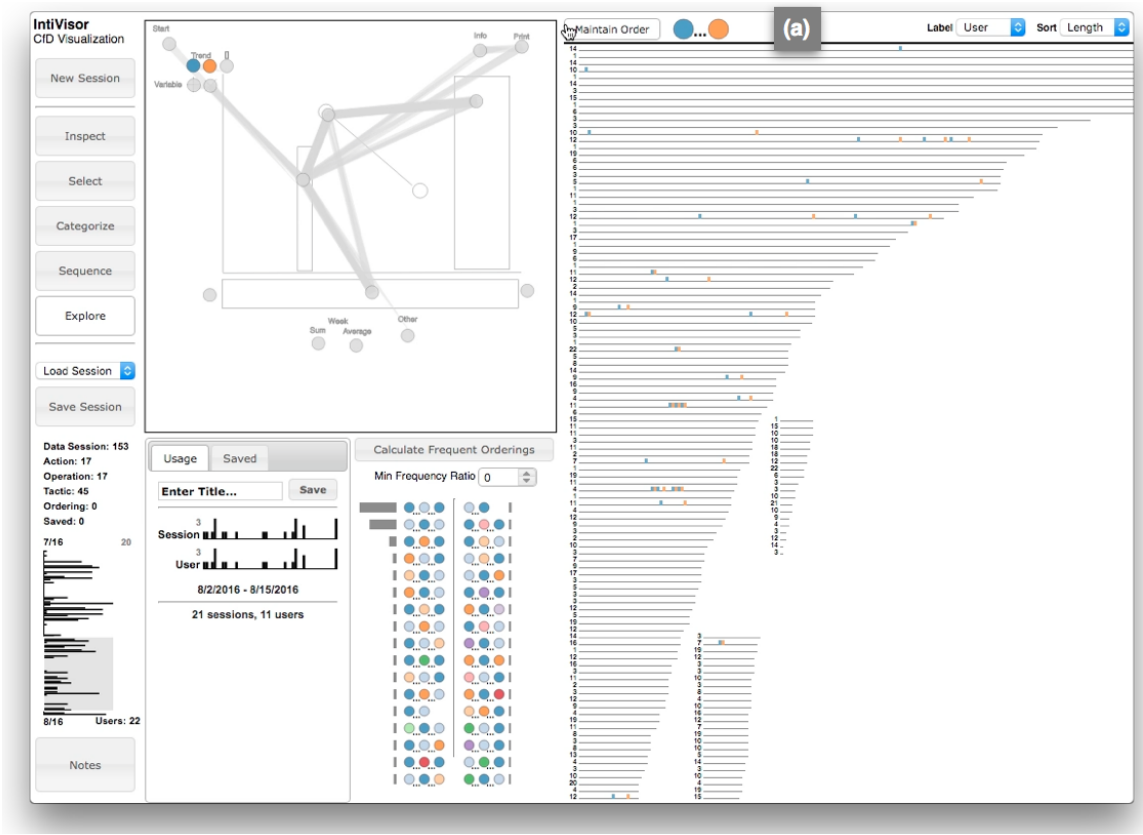


Figure 42: Selection example. (a) Selection area with two operations that occur in order.

An analyst may want to identify a set of operations only when they occur in a specific order, such as in the visual-information seeking mantra [41], but not necessarily back to back to discover higher-level analysis patterns. To find this type of patterns, an analyst can toggle on the "maintain order" mode in the selection. This toggle adds three dots between the selected operations and/or tactics, indicating that these items need to occur in this specific order but could have other operations between them (Figure 43b). The patterns are called "orderings."

- Consecutive Selection

  At times an analyst needs to find operations that occur consecutively to examine perhaps if a user is attempting to accomplish a specific task. To find this type of sequential pattern, an analyst can click on the three dots between the pair of items he/she wishes to be consecutive in the selection area. The three dots will disappear and the neighboring operations in the two items will be connected as partially overlapping circles, indicating no other operations can occur between the two items (Figure 43c).

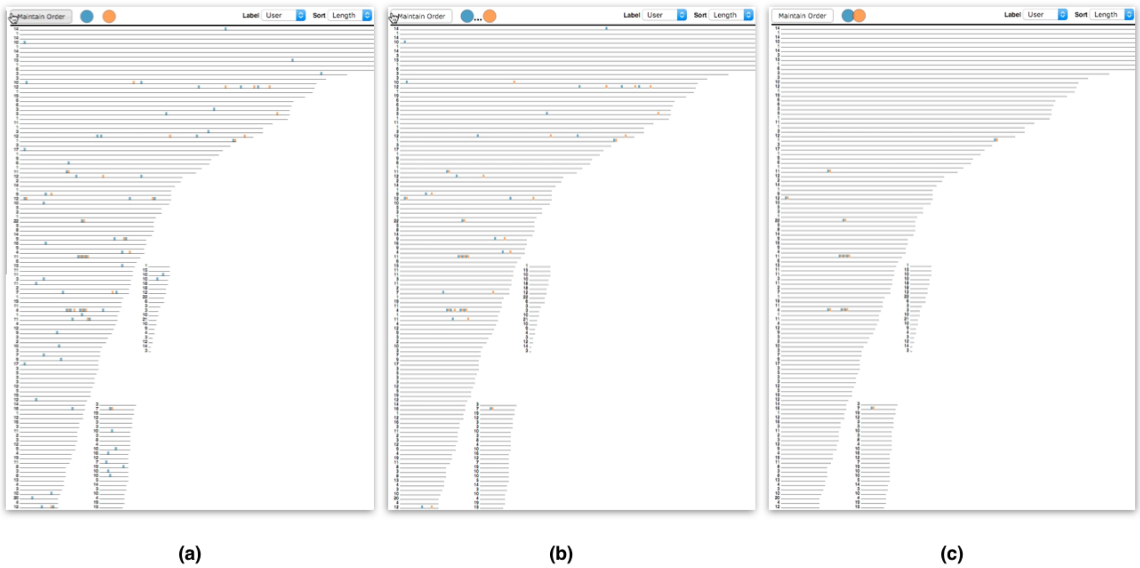


Figure 43: Three types of selections, using operations as example. (a) Independent selection. Each operation is selected independently. (b) Ordered selection. Only operations occurring in the selection order is selected. Three dots between the selected items indicate other operations could occur between the items. (c) Consecutive selection. The two partially overlapping circles indicate the operations have to occur consecutively to be selected. When a tactic is selected, it is also represented this way.

Using these selection mechanisms, an analyst can define his/her selections with a level of flexibility without the need to know regular expressions. This design trades off simplicity with complexity and is sufficient to form a large variety of useful patterns.

If an analyst wants to highlight one specific instance of an operation, he/she could indicate it in the bar view. One special option in the color selection menu of the bar view is a circle that allows highlighting an individual operation. Selecting this circle creates a circle around that individual operation and grays out all other operations (Figure 44). The corresponding operation in the graph view is also highlighted. This

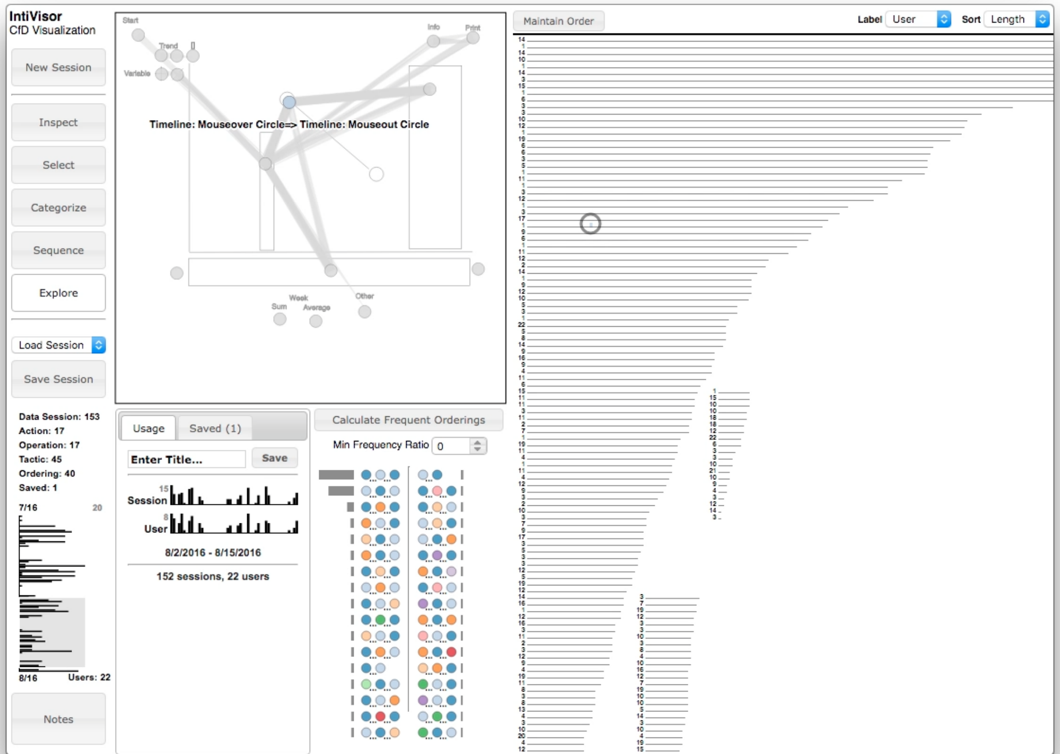


Figure 44: Highlighted individual operation: Timeline: Mouseover Circle→ Timeline: Mouseout Circle.

feature is different from the typical selection because it merely highlights that one individual operation and does not actually select it into the selection area, which would have highlighted all the occurrences of this operation in the bar view. If an analyst wants to figure out which operations occur before or after the highlighted operation, he/she could use the "left" and "right" keys to navigate through the operations. This feature essentially provides a method for an analyst to "walk through" a session, one operation at a time.

4.3.5.3 Usage Distribution View

The usage distribution view provides an analyst two charts for examining the session and user distributions over time. This view is located in the lower-left corner of the Explore view (Figure 33c). It includes two bar charts that display the usage distribution information over time. By default, it shows the occurrences of all the sessions in the bar view. When a specific usage pattern, such as a set of operations and tactics, is selected, the view is updated to show only the usage distribution of sessions that include the pattern.

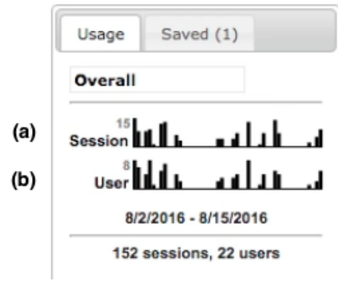


Figure 45: Usage distribution view. (a) Session distribution view. (b) User distribution view.

1. Session Distribution View

   This view is the bar chart at the top of the usage distribution view (Figure 45a). It shows the session distribution over time in a bar chart. The x-axis is time and the y-axis is the number of sessions. The time window is the same time range selected in the bar chart of the control panel. Each bar represents the total number of sessions within a given time block. The time block is determined by the time range of the x-axis divided over the available chart space. The y-axis is labeled with the maximum number of sessions in a given time block.

2. User Distribution View

   This view is the bar chart below the session distribution view (Figure 45b). It is constructed the same way as the session distribution view but with users instead of sessions. This view complements the session distribution view because in that view, when the number of sessions is high, it is not possible to know if the visualization application was used a large amount of times by a small number of users or was used a small amount of times by a large number of users. The user distribution view can help differentiate these these two cases because when the number of sessions is large but the number of users is small, it would be the first case.

4.3.5.4 Frequent Orderings View

This view shows the frequent orderings of operations in the data that might reveal higher-level usage patterns than tactics. The view is located in the lower-middle portion of the Explore view (Figure 33d). It lists the top frequent orderings of 2 and 3 operations in two columns. An ordering is a sequence of operations that do not necessarily occur back to back. The most frequent 17 orderings are listed in the left column and the next most frequent 17 orderings are listed in the right column. Each row in a given column includes two components: a visual representation of the ordering and its relative occurrence frequency to other frequent orderings. An ordering is visualized the same way as in the selections—circles with dots in between (Figure 43b). The circles, which represent operations, use the same default colors as those in the graph and bar views. The occurrence frequencies are visualized as horizontal bars next to the orderings. A longer bar indicates a higher relative occurrence frequency to the other most frequent orderings. The bars are to the left of the left column and to the right of the right column. The orderings can be filtered by a minimal frequency ratio, the same as in the Sequence view.

To minimize the extraction time of the frequent orderings for a better interactive experience, I used a sample. The sampling mechanism is different from that in the Select and Sequence views because the computational complexity of the algorithm is exponential in this case. The mechanism operates on two assumptions. First, long sessions may have different usage patterns from short sessions. Second, the first 200 operations of a session is sufficient to demonstrate the usage patterns of the entire session. As a result, the frequent ordering extraction only samples up to the first 200

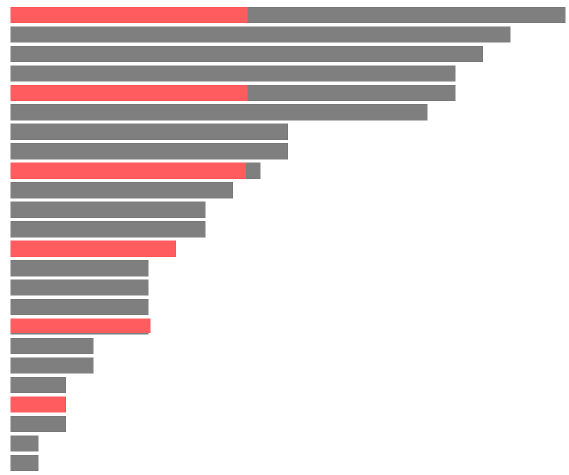


Figure 46: Sampling mechanisms for the frequent orderings. Each bar represents a session. Parts colored red are the sample. When sampling for frequent orderings in the Explore view, the first 200 operations of one out of n sessions are sampled when up to 2000 operations are included in the sample.

operations in one out of every n sessions from a list of length-sorted sessions. n is determined by maximizing the session coverage while sampling up to 2000 operations (Figure 46). The top frequent orderings are shown in Figure 33d. Moving the mouse cursor over each frequent ordering will highlight its operations in the graph view. Clicking on an ordering will select the pattern.

Using the Explore view, an analyst can flexibly combine the use of two complementing approaches for interaction analysis: (1) Find usage distributions from hypothesized usage patterns and (2) explore unexpected usage patterns from frequent occurrences of operations, tactics, and orderings.

4.4 Summary

Using IntiVisor, an analyst can flexibly organize events into actions, operations, tactics, and orderings to progressively reveal usage patterns at multiple level of granularity and abstractions from users of a visualization application. The system implements a key component in the visual interaction analysis framework step by step. In the next chapter, I will demonstrate the utility of the framework and IntiVisor from interactions collected from multiple field-deployed visualization applications.

CHAPTER V

EVALUATION

To evaluate the visual interaction analysis framework, I conducted a set of studies. The studies are designed to answer the research questions on understanding the flexibility and practicality of the framework in finding insights from the interaction logs of a visualization application. In this section, I first provide an overview of the studies and then present each study in detail.

Table 3: Overview of the design of the studies.

Study	Purpose	Format	Visualization Application	Duration
Participatory Design	Improve IntiVisor with potential users (analysts)	Multiple-session remote group workshops	CfD Visualization	1 hour per session
CfD Study	Assess individual perceptions on the flexibility, practicality, utility of framework	Single-session in-person individual studies	CfD Visualization	2-hour session
Field Study	Assess practicality in the field	Multiple-session remote group workshops	CfD Visualization	1 hour per session
Generalization Study	Generalize framework to multiple visualization applications	Single-session in-person/remote individual studies	CiteVis, VISLists, List View	2-hour session

Table 3 shows the four studies I conducted. Each study has a unique purpose in the evaluation process. The Participatory Design is a formative evaluation of IntiVisor in order to obtain feedback from potential users (analysts) for improving the system. The CfD Study is a summative evaluation of the framework for assessing individual perceptions on analyzing the interactions from the CfD visualization application (Appendix A). The Field Study is a follow-up of the CfD Study that is for assessing how the participants are able to use the system on their own in the following months. The Generalization Study assesses whether the framework is applicable to other visualization applications (CiteVis[1], VISLists[2], List View[3], Microsoft Research Data Visualization Apps for Office (Office Visualizations)[4]) to investigate the generalizability of the framework.

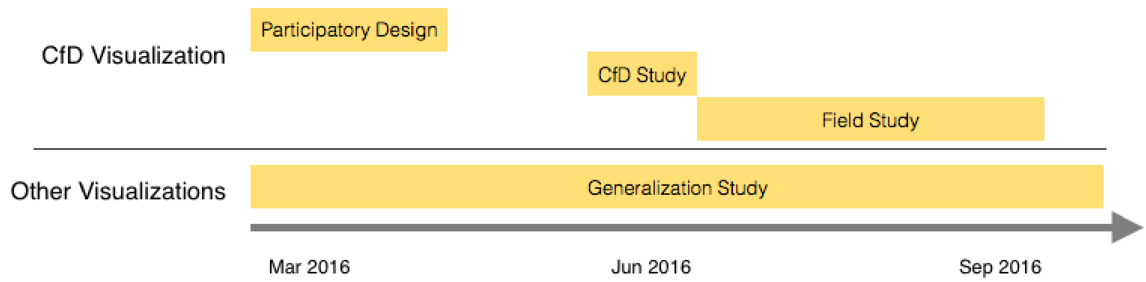


Figure 47: Overview of the evaluation process. This illustration is meant to show the relative temporal relationships of the studies so some study durations are intentionally increased to fit the labels.

Figure 47 shows a temporal relationship of the studies. The Participatory Design is conducted first using the CfD visualization interaction data to help improve IntiVisor. The CfD visualization was selected because at the time it already had about 1.5 years of interaction logs. At the same time, I worked with the designers of the other visualizations (CiteVis, VISLists, List View) in the Generalization Study to instrument their systems for collecting interaction logs. After the Participatory Design, a time window for updating IntiVisor was reserved before the CfD Study began. Following the CfD Study, the same participants were given access to the system for the Field Study. Meanwhile, when enough interaction logs from the other visualization applications were collected, I conducted the Generalization Study with designers of those visualization applications using the same study format as the CfD Study.

[1]CiteVis: http://www.cc.gatech.edu/gvu/ii/citevis/
[2]VISLists: http://www.iilabgt.org/vislists
[3]List View: http://www.iilabgt.org/listview
[4]Microsoft Research Data Visualization Apps for Office: https://www.microsoft.com/en-us/research/project/microsoft-research-data-visualization-apps-for-office/

5.1 *Participatory Design*

The Participatory Design study aimed to collect early feedback from users during the design phase through a set of workshops. Specifically, I sought to understand the participants' analysis needs, discovering limitations of IntiVisor, and finding new directions for improving the system. I conducted four design workshops remotely with the participants, each lasting for about an hour. Table 4 provides a summary of the workshop activities. In the first session, I was only able to give them an overview of IntiVisor. Later on, we were able to dig deeper into analysis with different views of the analysis system.

Table 4: Workshops in the Participatory Design study

Session	**Number of CfD Participants**	**Activity**
1	4	Walkthrough of all IntiVisor views
2	2	Detailed analysis of Inspect and Select views of IntiVisor
3	3	Reviewed CfD visualization version 1 and detailed analysis with all IntiVisor views
4	4	Detailed analysis with all IntiVisor views

5.1.1 Participants

Up to four CfD collaborators on the visualization application design team and I met remotely for the workshops. Of the five people in these study sessions, three of them, including me, were part of the original design team of the first version of the CfD Visualization. The other two CfD collaborators joined the team later to design a later version of the application. The CfD collaborators on the design team were at the management level so they were specifically interested in seeing whether certain key features were being used that may help assess their staff training efficiency.

5.1.2 Data

I included usage data from the first version of the CfD Visualization that was deployed in May 2014. This dataset only included an early version of the Timeline view. "Participatory Design" in Figure 48 illustrates the data subset for this phase of the study. This version of the visualization application was introduced to the users through a set of tutorial workshops in the beginning of the deployment. It was being actively used for about 6 months until a new version of the application (version 2) replaced it.

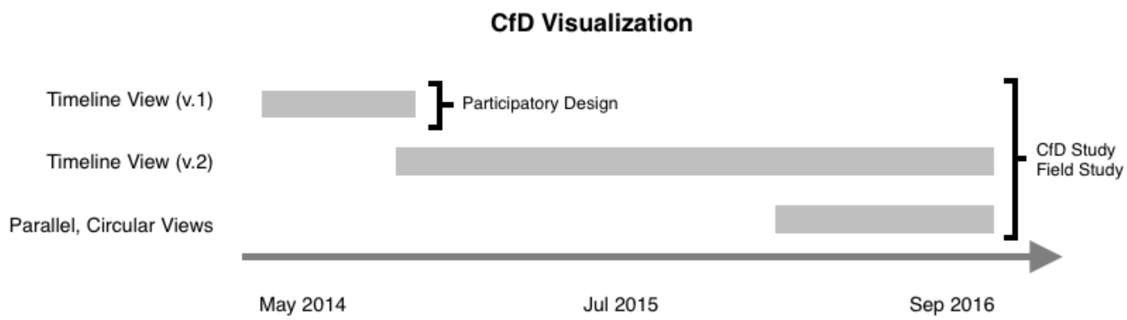


Figure 48: CfD visualization interaction data used in the studies.

5.1.3 Findings

The findings are organized into several categories: log data issues, interpretation issues, missing information, and missing analytic features. From the workshops, I learned and addressed a few limitations of IntiVisor on these issues.

Log Data Issues

As with any log data, potential quality issues such as missing log entries, inconsistent log content from updates, and duplicated log entries may be present. Being able to effectively identify and consider them in the analysis process can significantly reduce the noise in the data and improve the analyst's confidence in the insights found.

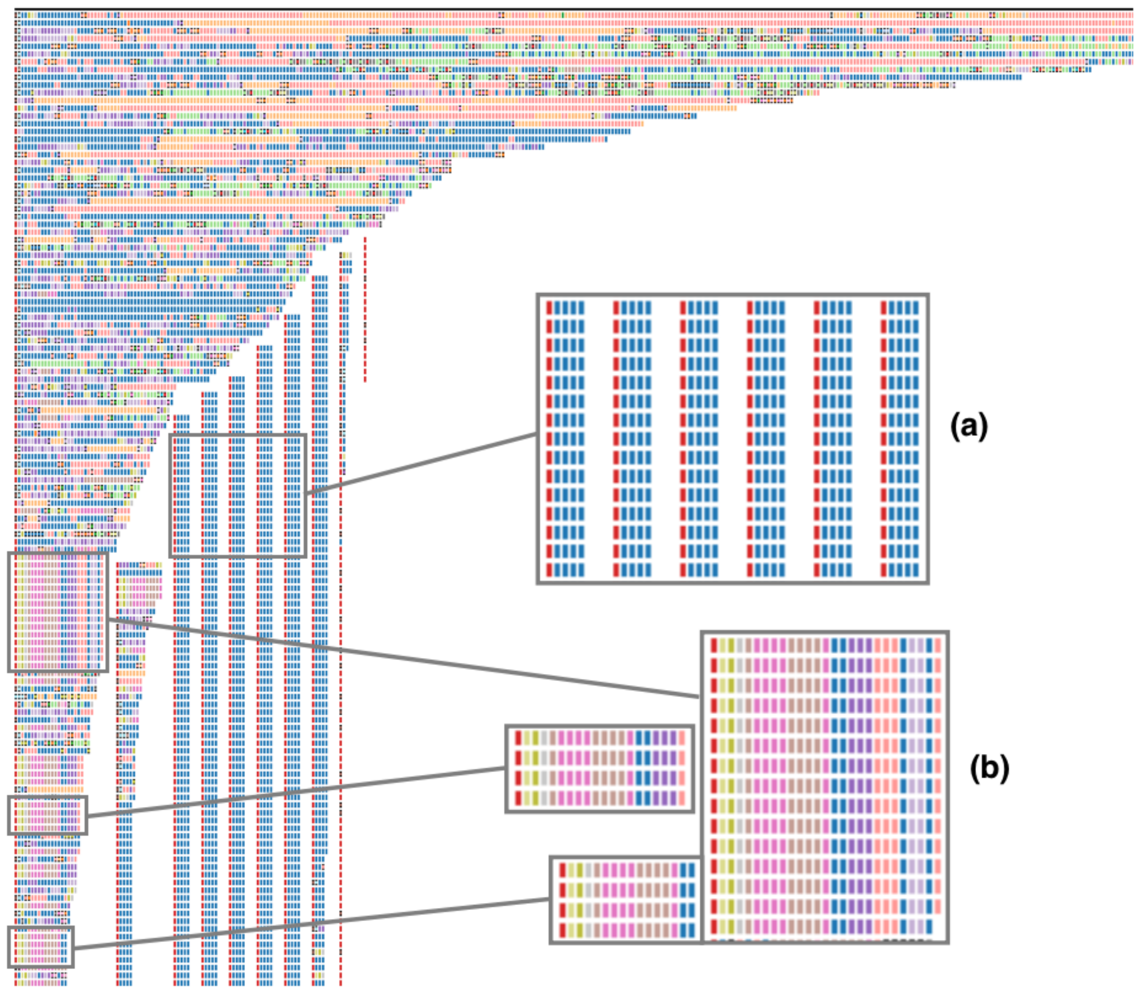


Figure 49: Duplicated sessions discovered in IntiVisor. (a) Duplicated sessions with exactly the same set of interactions. (b) Duplicated sessions that are progressively longer.

One log data issue that was present in the first two workshops is session duplication. From Figure 49, it is visually apparent that a large set of sessions have similar or the exact same set of user interactions. Upon further investigation, it turned out that a logging error may have occurred that caused a small number of sessions to be logged more than one time. At least one session seemed to become progressively longer in the duplicates (Figure 49b). These duplicates caused the emergence of sequential patterns that were only prevalent because they were present in the duplicated sessions. This issue could have been difficult to identify without a visualization.

If not identified, these patterns can mislead an analyst to believe that the patterns occurred significantly more frequently than they actually did. The visualization also provides a good overview of the prevalence of the issue and the variations of the duplicates. Using IntiVisor, I was able to remove the duplicates effectively by iteratively setting removal rules, applying the rules, and examining the removal outcomes until the problem was seemingly resolved. The duplicates were removed after the first two sessions.

Interpretation Issues

One of the biggest challenges of an analyst is to interpret the usage logs. For example, why is the frequency of a specific interaction higher than usual? The analyst may need to know about the design of the visualization application, the information in the usage log, or any other contextual information of the deployment environment. When some information is missing, an analyst's confidence in the interpretation could be significantly reduced.

When inspecting the frequency distribution of interaction events, it became apparent that interactions that set the default configurations were used less frequently. For example, the CfD visualization supports multiple types of data aggregation methods. The default data aggregation method is adding the data values up by month but the drop-down box that selects this configuration was the least used among the aggregation options. It is interesting because one might intuitively assume that configurations that were more frequently being set to were more frequently used. But in fact, the default configuration is often the most frequently used one because every user sees it at least once when the application starts. The interactions that led to default configurations were only for when the users selected a different configuration and then switched back to the default one. This realization prompted one of the participants to question whether the default configuration was "skewing" the data. I think this is a universal problem with interaction analysis when the state of the visualization application is inferred from the interaction sequence. The solution to the problem is to keep track of the default configurations of the visualization application when analyzing the data.

Sometimes a perceived issue is caused by unfamiliarity of the log data. For example, one participant observed a continuous string of “Click and Drag Time Range” events from a session in the Inspect view. This interaction occurs when a user clicks and drags a time window on an overview panel to zoom into the data in the main view. In an earlier version of the log, every value change during the dragging process was logged as a separate event. Therefore, during the dragging process, a long list of events were logged. The logging mechanism has since changed to only log an event when a user releases the drag to avoid the excessive amount of log data it generated. This mechanism was known to me who setup the logs but not known to the participants. Similarly, there was also a case when a log event was not recognizable by a participant because of a wording issue. These issues show how important it is in a log analysis process to know about how the logs were generated and mapped to the interactions.

The usage log showed that this version of the CfD visualization was not widely adopted. Most of the usages of this version (blue) were from the introductory tutorial sessions, as shown in Figure 50. Therefore, when less meaningful usage patterns emerged, it is likely caused by users trying things out in the tutorials. The participants were as a result less interested in over-interpreting these patterns because they were likely not realistic everyday usages. We actually have some data in this version that were everyday usages after the tutorial workshops. However, IntiVisor then did not have a feature to specify a subset of data to be used in the analysis. I therefore added a feature to support an analyst to select a time range of data for the analysis (Figure 17, session distribution chart).

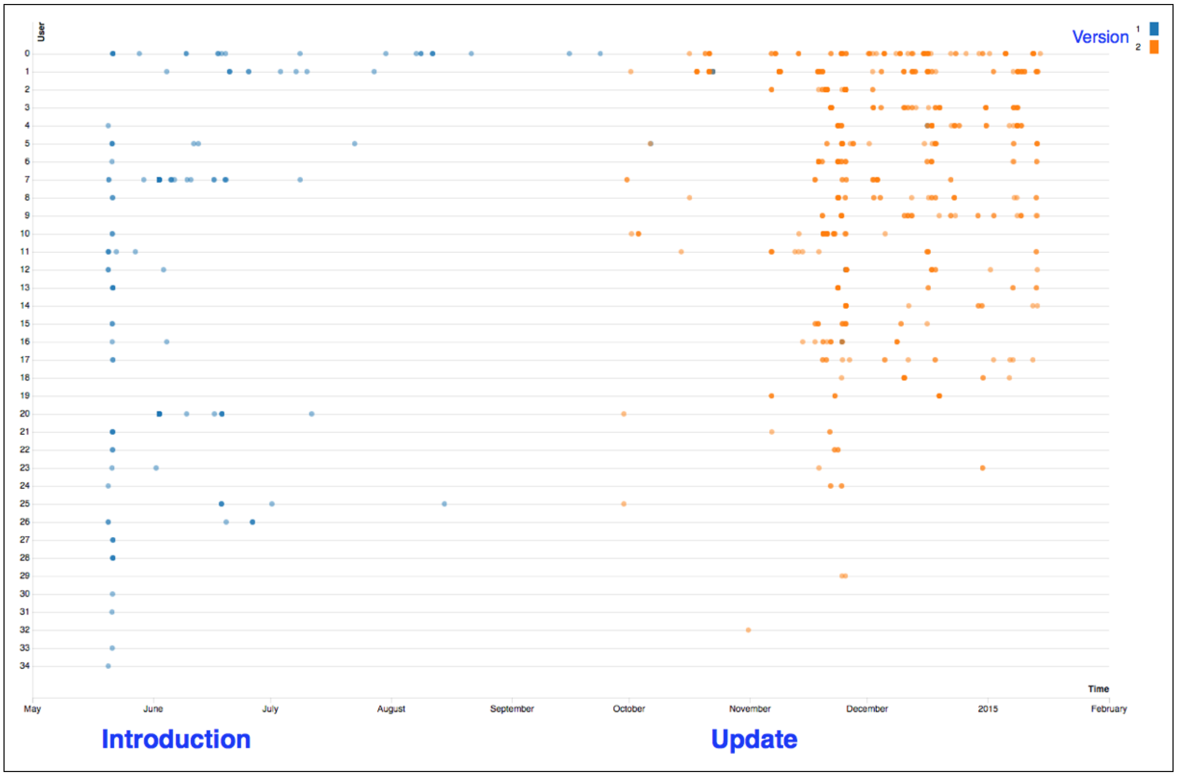


Figure 50: Usage of visualization tool by user over time. The users on the y-axis is ordered by the frequency of using the tool. Each colored circle represents a usage session. The colors are mapped to the two versions of the CfD visualization in about the first 6 months of deployment.

Missing Information

IntiVisor did not include some information of the usage data in all the views. From the workshop, I learned which of the missing information needs to be included and ideas on how it should be included in IntiVisor.

In the workshops, a key limitation of IntiVisor was that it cannot import all the parameters in an event. For example, in IntiVisor an event of aggregating data would be shown as “Drop-down Aggregation Selection” because a user needs to use a drop-down box to select the aggregation type. However, by default the aggregation type, which could be, for example, “Sum data by month” or “Average data by day,” was omitted in IntiVisor. This limitation was significant as the participants were interested in seeing how the different visual configurations were used. To address this issue, starting in the third session, certain key parameters such as the aggregation type were added to the events in IntiVisor. After the workshops, a new feature was added to support including parameters in events on demand (Figure 23, on page 50). An analyst can find a list of available parameters and select which ones to include in an event.

The participants were particularly interested in using IntiVisor to assess the training progress of their staff members, who are users of the CfD visualization. Therefore, being able to examine whether the staff members were able to use the key features of the CfD visualization is important. However, the only place where an analyst could find user information was in the Inspect view. As a result, I added the user information to the Explore view in a later version so that an analyst could examine and compare usage patterns within and between users at a later stage of the analysis.

One specific request from the participants is the inclusion of usage reasons of the CfD visualization in IntiVisor. In the CfD visualization, whenever a user accesses the system, he/she is prompted to select/enter a reason, such as team meeting, for using it. The participants were very interested in seeing how the application was used for different reasons. This information was not included in the version of IntiVisor used in the workshops but was added later.

Missing Analytic Features

IntiVisor supports a wide range of analytic features but sometimes an analyst may need a feature that is not supported. IntiVisor cannot support all popular analysis methods but should include those that are important to the study participants. For example, one participant was interested finding which interactions led to a bookmark. Unfortunately, IntiVisor did not have a feature to easily extract this information. As a result, in a later version, I added a feature that can find interaction sequences not only leading to a specific interaction, such as bookmarking, but also following a specific interaction to see what occurred next (Figure 29).

5.1.4 Discussion

From the Participatory Design, I learned about many limitations and opportunities to turn IntiVisor into a more useful system. For example, several pieces of information in the usage log, such as event parameters, user, and usage reason, were not loaded in relevant views in IntiVisor. When some relevant information is missing, the problem could significantly impact the interpretation. I also learned about other features to add, such as selecting a data subset and finding which events occur before or after a specific event, that would be informative to an analyst.

Second, I obtained an understanding on how much knowledge was required to interpret interaction data. Some participants were on the design team of this version of the CfD visualization. But since this version of the application was deployed, used, and replaced over a year ago, our knowledge about the features available back then were quite fuzzy and unreliable. Therefore, we had to revisit that version of the application and reference the change logs during the analysis process. The change log needs to not only be of the visualization application but also of the interaction log. Another issue is that I instrumented the CfD visualization to log interactions. Therefore, the participants had little knowledge about the log data. They did not know exactly which interactions were logged and how they were logged. Luckily, a large amount of this information was self-explanatory in IntiVisor. But occasionally the participants would still need some help in clarifying this information. If an analyst did not have access to this information, the interpretation could be significantly more difficult. Third, an analyst needs to know how to use the log analysis system, in this case, IntiVisor. IntiVisor is an expert analysis system that requires a thorough tutorial for someone to learn how to use it. The learning curve makes the system only for those that are interested enough in analyzing the data in depth. I can, however, reduce this learning curve as much as possible by increasing the ease of use of the visual interaction analysis system.

Overall, the Participatory Design successfully accomplished its goal in assembling a list of new features that could be added to IntiVisor. The details of these features were described in Chapter 4. After these features were added, the remaining studies were conducted.

5.2 CfD Study

This study is the first summative evaluation study. I met a number CfD staff members who were on the design team of the CfD visualization individually to observe and interview them about their experience using the framework.

5.2.1 Participants

The same four participants from the Participatory Design participated in this study. They were CfD staff members that were on the visualization application design team. Because they were the same participants, this study also included usage data from the most recent version of the visualization application. As a result, the participants were able to focus their analysis to a completely different set of data from that used in the Participatory Design.

5.2.2 Data

Different from the Participatory Design, this study included all the log data from both versions of the CfD visualization. That includes more than 2 years of data, as shown in Figure 48. I instrumented the application to log its interactions. Over time, the visualization application evolved and so did the interaction log. For the study, all interaction data were available to query directly from the database but only the most recent two weeks of data were visualized by default.

5.2.3 Process

I met each participant individually for a two-hour study session (Table 3). I first walked through the features of the most up-to-date version of IntiVisor. Next, I let the participants operate the system by themselves while thinking aloud with their analysis process. If a participant was not comfortable operating the system, I would help operate it for him/her but the analysis process would still be driven by the participant.

During the analysis, when a specific pattern seemed noteworthy, I would prompt the participant to save it and answer five questions about the pattern (Table 5). The five questions support answering research questions 2 and 3. After the analysis session, I conducted an interview (Table 6). The interview questions are designed to support answering research questions 1 and 3. Additionally, I logged the states of IntiVisor, which for example include the actions selected and tactics extracted, whenever a participant navigated to a new view or saved the session.

Table 5: Questionnaire for each saved patterns

Question	Response	Support
What is the level of knowledge you have about this pattern's existence?	Low/Medium/High	RQ2
What is the level of unexpectedness of this patterns usage?	Low/Medium/High	RQ2
Does this pattern indicate a design issue?	Yes/Maybe/No	RQ2
What is the level of insight gained from this pattern?	Low/Medium/High	RQ2
What is the level of effort in finding the pattern and insight?	Low/Medium/High	RQ3

5.2.4 Patterns

I present patterns explored and/or saved by the participants. Saved patterns include the questionnaire response. In this section, I refrain from interpreting the data and

Table 6: Interview questions

Question	**Support**
Are there patterns that you wanted to find but could not specify with the system?	RQ1
What was the most laborious aspect of the analysis process?	RQ3
Were there any steps in the analysis that you think were too laborious for the process to be practical? If there were, what were they and why were they too laborious?	RQ3
What do you think are the most useful features of the system?	–
What do you think are the challenges in using and adopting the system in your work?	RQ3

leave that to the Discussion section that will be presented next.

Table 7 lists all the patterns that have been explored by the participants. Participants may explore the usage of one single interaction event, such as "Move annotation label." They may also explore interaction sequences. Sequences with consecutive interactions are connected by "→." For example, "Start → Remove (behavior)" is a sequence indicating that users started from removing a behavior in the visualization application. On the contrary, sequences with non-consecutive interactions are connected by "...". For example, "Toggle shift2 ... Print" indicates that a user inspected the night shift data before printing, but other interactions could occur in between.

Table 7: List of patterns explored in the CfD Study. Patterns connected with "→" indicates that the interactions before and after the "→" need to occur back to back. Patterns connected with "..." only needs the interactions before and after the "..." to occur in that order but not necessarily back to back. The amount of use by session and user of each pattern are listed in the corresponding columns. In this table, all interactions in the visualization view occurred in the Timeline view.

CfD Study
Number of Users: 23
Number of Sessions: 356
7/16/16-8/16/16

Pattern	Length	Session	User
Toggle shift2	1	122	11
Average data by month	1	26	9
Average data by week	1	26	7
Sum data by month	1	69	12
Sum data by week	1	24	8
Add sleep	1	12	4
Add SCIP	1	10	4
Toggle LOA	1	30	7
Toggle no data recorded	1	32	9
Add bookmark	1	6	4
Save bookmark	1	5	4
Toggle distributed category	1	11	6
Remove behavior	1	135	19
Remove medication	1	103	12
Drag and drop workbook	1	355	23
Toggle Trend	1	105	20
Add annotations	1	43	13
Print	1	176	20
Move annotation label	1	26	7
Drop-down bar selection (line)	1	43	9
Drag and drop bookmark	1	15	6
Start -> Remove behavior	2	12	4
Start -> Remove medication	2	5	3
Shift2 -> Sum data by month	2	9	2
Print -> Remove behavior	2	6	2
Show shift2 (show) -> Hide shift2	2	24	6
Remove behavior-> Drop-down bar selection (line)	2	2	2
Drop-down bar selection (intensity) -> Drop-down bar selection (duration)	2	32	7
Show trend -> Toggle trend (zoomed) -> Toggle trend (all)	3	9	4
Show trend -> Toggle trend (zoomed) -> Print -> Add sleep -> Toggle trend (all) -> Print	6	0	0
Mouseover bar (Aggression)...Mouseover bar (Sleep)	2	1	1
Toggle shift2...Print	2	82	10
Toggle (Text/List)...Add bookmark	3	2	1

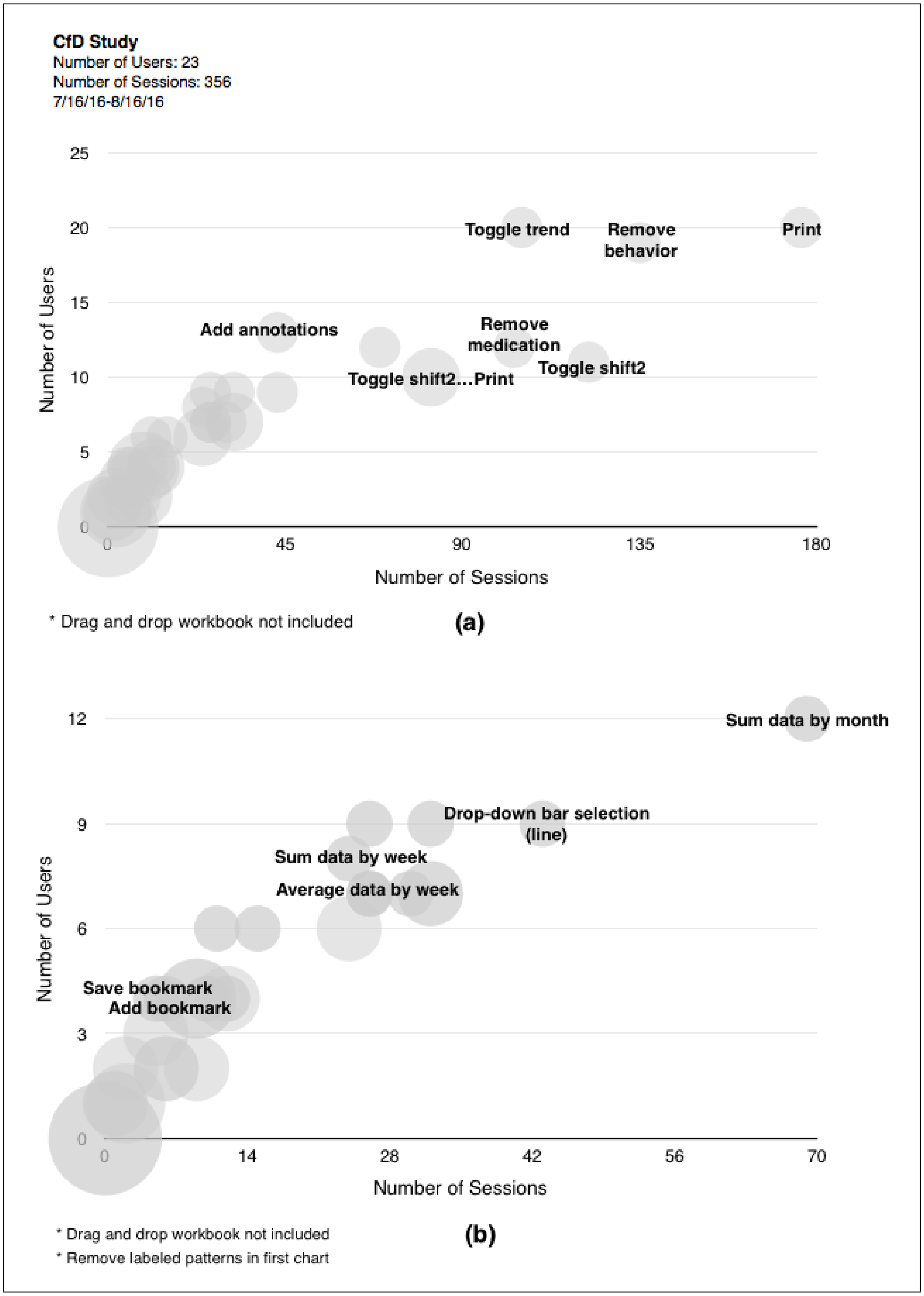


Figure 51: The amount of users and sessions of the explored Interaction patterns in the CfD Study. The size of the circles are mapped to the length of the interaction sequence, disregarding whether the items in the sequence occurred back to back. (a) All patterns except "Drag and drop workbook" because of its prevalence. (b) Subset of patterns without the labeled items in (a).

The amount of usage of these patterns could indicate whether they are a well-adopted analysis method or simply a barely used feature. Specifically, differentiating the amount of users and sessions can help determine whether a popular pattern is used extensively by a small number of users or occasionally by a large number of users. To identify the amount of users and sessions for each pattern, I extracted two weeks of data from the interaction log that included usages from 23 users and 356 sessions. I used the amount of users and sessions as the two axes of a scatterplot to map the interaction patterns (Figure 51). Each circle represents a pattern. Due to the amount of overlapping of the patterns, I only labeled the outliers in Figure 51a. For a closer look, I created another chart without the labeled outliers in Figure 51a and rescaled the axes. The second chart is labeled with its outliers, as shown in Figure 51b. The size of the circles are mapped to the number of items in the interaction pattern, disregarding whether the items occurred consecutively. For example, the circle for Start → Remove (behavior) is of the same size as the circle for "Toggle shift2... Print."

A subset of explored patterns are saved by participants. To save a pattern, a participant needs to answer the questionnaire of this study that includes five multiple choice questions as listed in Table 5. The saved patterns and questionnaire responses are listed in Table 8b and Table 9. The participants each saved 2-6 patterns. The titles of the saved patterns are given by the participants. The responses to the five questions are color-coded by the method in Table 8a. Using a traffic light metaphor, a interesting response to the study is colored green, such as having a low knowledge of existence of a pattern. This response is interesting because it indicates the framework helped discover unknown patterns. On the contrary, a uninteresting response is colored red, such as no indication of a design issue. All responses are on a three-point scale so the middle responses are all colored yellow. Each response is mapped to a number (Table 8a). The average number of a given question is listed at the bottom of Table 8b.

Table 8b and Table 9 organized the patterns differently. Table 8b aligns the questionnaire responses so that response values could be easily compared between patterns and participants. Table 9 places patterns in a different orientation. Patterns that are similar are highlighted with the same background color and placed in the same column. For example, participants 1, 2 and, 3 all explored the use of night shift data (blue background).

5.2.5 Discussion

Based on the patterns explored, in this section I discuss how they can answer the three research questions.

A. How do we provide flexibility in the analysis process for identifying patterns?

To determine whether the framework was able to help an analyst flexibly organize events to discover a wide range of interaction patterns, I present the following chart to organize the findings.

Variation of Patterns

An interaction pattern discovered can be organized along two dimensions: composition and abstraction (Figure 52). This chart is related to the Task Cube designed by Rind et al. for organizing tasks but without the "Why" dimension [36]. Composition reflects how many interaction events are within a pattern. Abstraction reflects how generalizable a pattern is. A logged event should include little composition and abstraction and be located in the lower-left corner of this chart. For example, the "Click Zoom In button" event in Figure 52 is located there. Moving along the Composition axis, the event is grouped with other interaction events such as "Click Zoom Out button" and "Click Pan Left button," to form more complex patterns. On the other hand, moving along the Abstraction axis, an event is "abstracted" or "renamed" into a more generic category. For example, "Click Zoom In button" in Figure 52 can be renamed into more generic categories "Zoom" and "View port change." When events are both composed and abstracted, they are grouped and renamed into a new unit. For example, "Click Zoom In button → Click Zoom Out button" can be grouped and renamed into "Zoom adjustment." An analyst often composes and abstracts events to organize events into a smaller, more meaningful set of categories during the analysis (e.g., [13, 34]). This process means converting events from the lower-left corner into patterns in the upper-right corner.

IntiVisor supports converting events into patterns further along the two dimensions in multiple views. Composition is supported by the Select, Sequence, and Explore views. In the Select view, an analyst can select pairs or triples of events as a new unit, action. These selections are patterns with a higher level of composition. In the Sequence view, frequent sequences of consecutive operations are automatically extracted. These sequential patterns further compose the actions into tactics which have a higher position in the composition axis. In the Explore view, two methods are available for composing events. An analyst can manually compose events into patterns from selections or use patterns automatically extracted as frequent orderings.

On the other hand, abstraction is supported in the Categorize and Explore views. In the Categorize view, actions are mapped to a position on a canvas. The canvas includes a drawn context that defines a set of categories as a perspective. This perspective allows an analyst to categorize one or more actions into an operation. The categorization can be an abstraction process when multiple actions are mapped to the same operation, essentially renaming them into a new, more abstracted unit. In the Explore view, the abstraction is conducted by assigning the same color to different operations during pattern selection. In this case, the selected operations will seem to be the same in the views because they have the same appearance, visually renaming them into the same new abstracted unit.

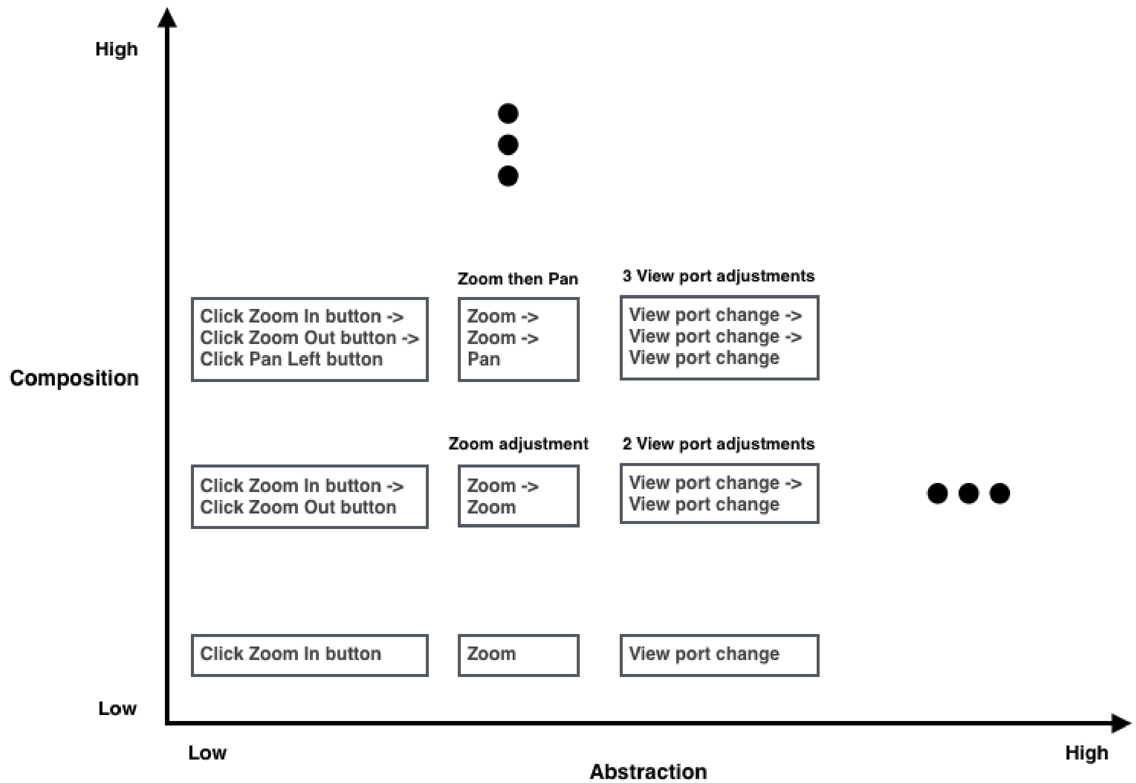


Figure 52: Organizing interaction patterns long two dimensions: composition and abstraction.

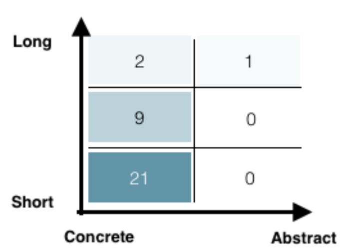


Figure 53: Organizing the patterns discovered from CfD Study of the CfD visualization. The numbers in the chart areas show how many patterns are within the corresponding spaces.

To demonstrate flexibility in event organization, an interaction analysis system needs to support its analysts in organizing events into all the areas of the chart in Figure 52. As shown in Figure 53, I mapped the patterns discovered from the system for this study into six areas in the chart. The y-axis is divided into three groups (rows). The bottom row is for single events; the middle row is for pairs of events; and the top row is for patterns with more than two events. The x-axis is divided into two columns. The first column is for single events and the second column is for any events or event groups that have any type of abstraction. The reason why I only chose to use two levels in the Abstraction dimension is because it is harder to determine the level of abstraction applied by an analyst. The number of patterns are labeled within each area. Each area is color-coded with the relative percentage of the pattern (higher percentage, darker shade). From the figure, it is clear that the majority of the explorations were focused on individual interaction events. This observation is probably because individual interactions are the easiest to interpret. Nevertheless, some participants were able to extend the exploration into composed, sequential events and occasionally abstracted representations of events. This first study shows that the system was able to flexibly support the discovery of composed patterns but not yet for encouraging the exploration of abstracted patterns.

Variation of Perspectives

Another piece of flexibility is demonstrated through the perspective defined in the Categorize view. An analyst must be able to flexibly define a perspective by a set of categories to support his/her analysis needs. From the studies, the participants defined different perspectives, showing the flexibility of the system in this aspect. Figure 54 shows the perspectives used by the participants. Participant 1 carefully mapped actions to four separate regions: ADD VARIABLE, ADD ANNOTATION, BOOKMARK, AND PRINT (Figure 54a). This mapping was used because this participant was particularly interested in learning about which activities led to annotations, bookmarks, and prints, which may indicate findings in the system. After assigning a set of actions, the participants proceeded to use the Grid Layout feature to automatically place the remaining actions in a grid. But unfortunately, some automated action placements were too close to the existing operations on the canvas; therefore, the participant further rearranged the assigned actions. The entire process took about 20 minutes.

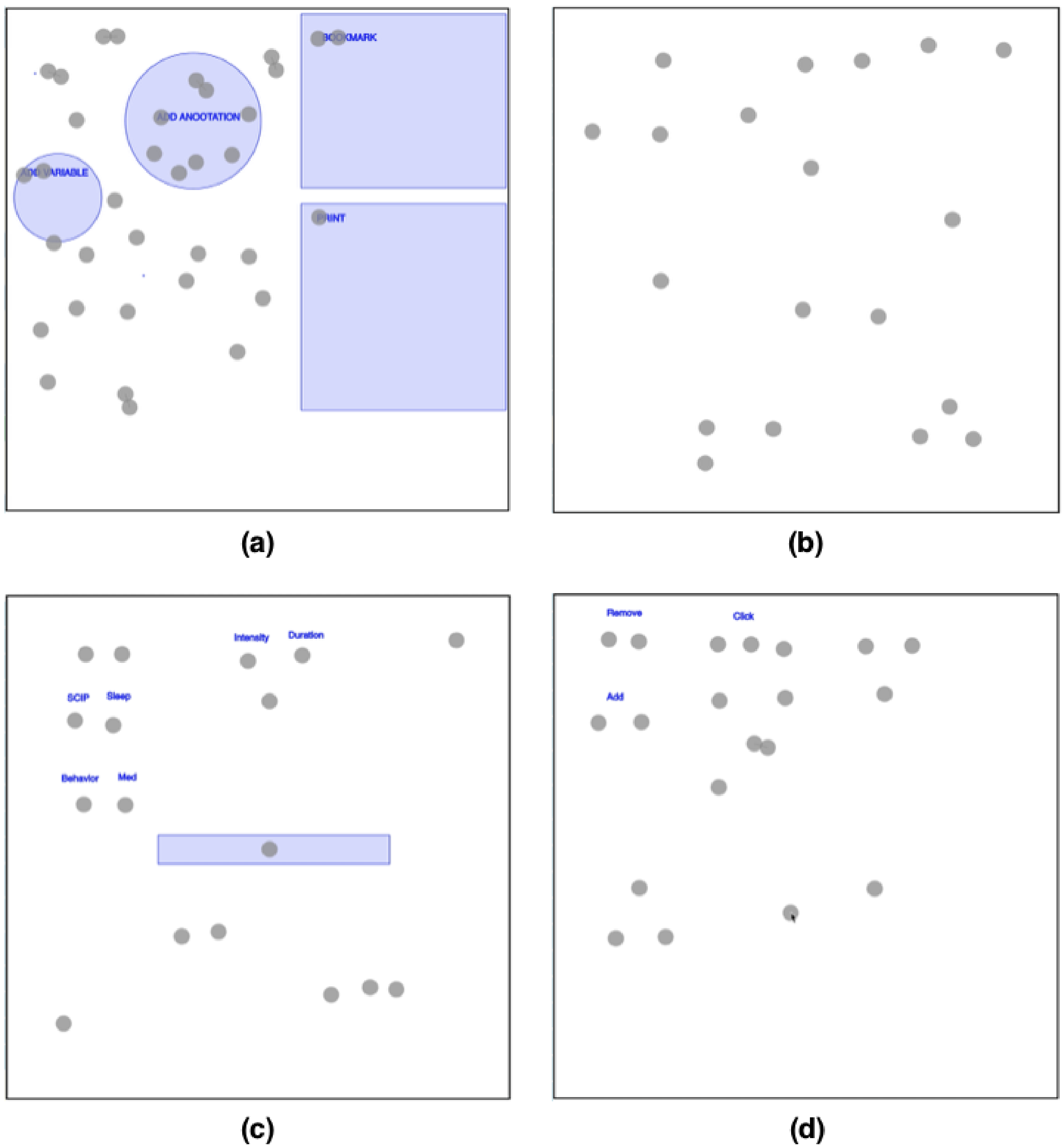


Figure 54: Perspectives defined in the CfD Study. The figures (a)-(d) map to the perspectives used by participants 1-4.

The perspective used in participant 2's study is the opposite of that of participant 1's (Figure 54b). All the actions selected by the participant were initially automatically assigned as operations in a grid layout. No drawings were created in the perspective so the step merely took a click to map the selected set of actions onto the canvas. As a result, the participant needed to rely on the interactions in IntiVisor that toggle on labels in the other views to identify individual operations. After further explorations, the participant discovered other interactions that might be of interest. Those new interactions were added to the bottom, blank area of the canvas for easy identification.

The perspectives used in participant 3's and 4's studies were similar (Figure 54cd). They both started with a simple, manually created layout that consists mostly of labels. The relevant actions were organized around these labels. This organization method required minimal drawings from the participant to create a meaningful context for the assigned actions. But even through the two participants used the same categorization method, the perspectives they defined were still quite different. This difference is evident from the labels they used. No labels were the same even though some actions were. For example, participant 3 chose to use the variable labels, such as SCIP, Sleep, Behavior, and Med to indicate which ones of them were added to or removed from the visualization application whereas participant 4 simply used the labels "Add" and "Remove" to indicate the adding or removing of any variables or annotations. Participant 4 also used a label "Click" in the perspective that organizes all the interactions that were from clicking a mouse button. These organization differences from the participants show how IntiVisor was able to flexibly support their varying preferences and needs.

Participants 1 and 4 categorized multiple actions into single operations which are shown as connected circles in Figure 54a and Figure 54d. As mentioned earlier, this categorization implicitly renames those actions into a single new unit. The new unit would be more abstract as it includes more than one original interaction. This feature supports the flexible discovery of patterns that move along the abstraction axis in Figure 52.

Pattern Discoverability

All the participants indicated that they were able to find all the patterns they looked for in IntiVisor from the interviews after the study. One participant mentioned that it was mostly because I was available to help in the discovery process. However, being able to identify all the patterns that all the participants were looking for shows the flexibility of the system in supporting the analysis process.

In summary, the flexibility of the framework is demonstrated from the variety of patterns explored along the two dimensions in Figure 52, the variety of perspectives generated, and that participants were able to find all the patterns they were looking for. Although the patterns were short on abstractions, I think it was mostly because the participants were still unfamiliar with the system and interaction analysis. In the next study, I will show that after being prompted to look for higher-level patterns, they were able to further the explorations into more abstract patterns.

B. Which types of insights can analysts gain from the identified patterns?

Many insights were discovered from the study. An insight is typically discovered when a pattern's usage amount does not map to an analyst's expectation. To illustrate these insights, I refer to Figure 51 that shows the user and session amounts of the explored patterns.

In Figure 51, the amount of users should be positively correlated with the amount of sessions for an explored pattern. As a result, the outliers in the chart, which are patterns that are off the diagonal trend line, are displaying an unbalanced user/session amount. The obvious outlier is Print. But the excessive use of Print is not considered a surprising finding because the CfD has a practice of using paper charts. Similarly, the extensive use of trend lines, where most users employed them in about 1/3 of sessions, is not surprising because inspecting trends is an important data analysis task.

However, some other outliers, such as Toggle shift2 (night shift), are more interesting. The usage chart indicates about half the users toggled on the night shift data for inspection. The CfD staff has three work shifts, two day shifts and one night shift. The CfD visualization has a function to optionally display data from specific work shifts. By default, the night shift data were not displayed in the visualization application because the CfD collaborators determined that it might not be as important as the data collected in the day shifts. However, based on the usages, it seems that about half of the users still preferred to include night shift data in their analysis. This observation was considered an insight from three of the participants (Table 8b and Table 9, column 1). One participant mentioned that perhaps the use of night shift data was related to the reliability of the staff recording data. Some night shift staff do a better job at recording data than the day shift staff. Therefore, it might be one reason why including the night shift data was one of the most used features in the application. But if the night shift data was so useful, one question arises from this observation: should the night shift data be included by default then? But taking a closer look at the amount of sessions that included the night shift data (122) (Table 7), it was still much less than 50% of the number of usage sessions (356) in the dataset. Therefore, the CfD design team still decided not to include night shift data by default.

A significant amount of data removal operations were discovered in the interaction data. Two participants saved this pattern and indicated it as an insight (Table 8b and Table 9, column 3). But one of the participants did not find the usage amount unexpected. This participant offered an explanation for the medication removals: many of them might be "discontinued" so users may opt to remove them to reduce these likely irrelevant data in the chart. However, nearly twice as many users removed behaviors. Behaviors do not have the "discontinuation" problem as medications, so the users must be removing them for another reason. One potential reason is related to a known data quality issue where sometimes additional behavioral variables that carry invalid data were incorrectly displayed. As a result, a user might start the analysis session by removing these erroneous data with this interaction. But from inspecting the log, I can tell at least about 70% of removals (370/527) were not because of this issue. Therefore, at least 70% of sessions included removing behaviors for other reasons, perhaps also as simple as cleaning out less important behaviors for the analysis at the time.

Removing the labeled outliers from Figure 51a, I rescaled the chart into that in Figure 51b to take a closer look at that subset of patterns. Examining the outliers on the extreme ends, several findings emerged.

The pattern with the most usage is "Sum data by month," a data aggregation feature in the CfD visualization. This feature adds up the data values within each month and shows the aggregated amounts by month. What is interesting about the extensive usage of this feature is that this aggregation method formerly was the default aggregation method in the visualization application. The problem with summing data is that a month of data could have 20 data points recorded or 30 data points recorded because of a variety of reasons, such as a "leave of absence." Sums of 20 data points cannot be directly compared with sums of 30 data points. As a result, we changed the default aggregation method to "Average data by month," which accounts for the amount of data points available. However, interestingly, many users chose to go back to selecting the previous aggregation method. Two participants were surprised by the amount of usage for this pattern so they saved it (Table 8b and Table 9, column 2).

One of the participants was uncomfortable about this finding because of the downsides of using sums. We even had a discussion about potentially including a warning in the application if a user chooses to select this option when there are many missing data points. However, the other participant was not as concerned. The participant provided a reasonable rationale for using sums: when the aggregated data value is low. For example, some behaviors only occur a few times in a month. If averaged, the aggregated value could be very low and thus less visible on the visualization. As a result, changing the aggregation method to sum can overcome this issue. In summary, this pattern was useful to discover because it helped the participants, who were on the CfD visualization design team, reflect on their designs based on the usages within the visualization application.

On the other end of the chart, bookmarks are one of the least used features in this visualization application. In the application, a user can add a bookmark to keep the view's configuration for later reference in the session. If a user wishes to use a bookmark in a later session, he/she could save the bookmark and import it later in sessions that need it. As part of a research study, when a user saves a bookmark, a questionnaire is displayed that requires the user to fill out before the bookmark can be saved (Figure 78). At the design stage, I envisioned the bookmarks to be one of the most used features because the CfD users can create and save them during a meeting preparation and then reload them during the actual meetings. However, from the log data, only a small portion of sessions were actually used within meetings (e.g., in one study, 3/342 sessions in two weeks). Users were probably extensively using printed paper charts for the meetings, leaving one important use case of the bookmarking feature out of the data. One participant later speculated in the Field Study that the lack of bookmark use could be because the limited need to revisit previously configured views. In other words, users want to start fresh as a previous analysis may be outdated.

An alternative explanation is that the bookmark feature was too difficult to use. Adding, saving, and loading a bookmark was not simple in the application. To keep a visualization configuration, a user first needs to add a bookmark. Next, the user needs to save it as a file. If a bookmark was not added first, the saved file will not include a bookmark. Afterwards, when the user wants to reload this bookmark in another session, he/she needs to find it in the file system and then load it with the corresponding workbook data. Afterwards, the user can find the loaded bookmarks in the Bookmarks tab. The entire process was not as streamlined as it could have been. Furthermore, saving a bookmark required a user to fill out a questionnaire that likely reduced the motivation to save it (Figure 78). Therefore, we decided to give a workshop about this feature after this study to remind users about how this feature works. We also considered reworking the feature to reduce the effort in using it.

In summary, in this first study, it is apparent that the participants largely focused on exploring individual interactions instead of sequences of interactions (Figure 53). However, even from individual sequences, they were able to identify insights based on their own expected amount of usage and their own knowledge about the data and the visualization application.

C. How do we provide analysts the capability to practically identify and analyze patterns?

The overall effort rating for identifying the saved patterns were 1.83 on a 3-point (1-3) scale with 1 being of the lowest effort (Table 8b). This high average rating shows how the participants were having difficulties using the system. However, since the patterns listed in Table 8b were listed chronologically by the time they were saved for each participant, the participants overall did seem to think the effort for identifying patterns to be lower as the study went on, indicating the higher efforts could have been because of the initial learning curve.

To further assess the effort required to use the framework, I interviewed the participants. Three interview questions asked participants about which aspects of IntiVisor did they find laborious during the study. The goal is to minimize the effort in learning, using, and adopting the framework for it to be practical.

Two participants considered the entire process of learning and memorizing the features in IntiVisor as the most difficult aspect. IntiVisor introduces a completely new set of data analysis tasks and tools to the participants who did not typically analyze interaction data. Even though all the participants were part of the Participatory Design, they still have trouble managing the features. As a result, during the study, I often needed to take over the operation of the system to help overcome the learning curve. One participant specifically indicated that if I was not assisting with the analysis process, it would be difficult to remember how each feature works. A system as complicated as IntiVisor has a learning curve that needs to be further reduced for this population.

One participant considered the event selection process to be the most laborious aspect. The CfD visualization has many features so the inspection of the entire list of interactions being operated by users can be laborious. Furthermore, since I instrumented the visualization application to collect its interaction logs, the labels of the logged events were sometimes difficult for the participants to intuitively map to the actual events. The participant suggested that perhaps by default abstracting the interactions into a smaller set of categories may help reduce the effort in this stage. The challenge is that this level of abstraction will need to be made at the logging instrumentation stage. For example, if all the zooming interactions, such as zoom in/out, should by default be abstracted into a "Zoom" category, this information will need to be recorded in the log entry. Afterwards in the system, a feature will need to be available to "unpack" the category if an analyst wishes to differentiate Zoom in from Zoom out. He/she needs to have a way to "extract" these interactions, perhaps using a feature similar to that for including parameters.

5.2.6 Conclusion

I found that the framework seems to reasonably support the flexible event organization and pattern discovery process, help identify insights, but was not easy enough to learn and use to be practical for this population. After this study, I provided IntiVisor to the participants for the Field Study to see what they could learn by themselves over a longer period of time.

5.3 Field Study

The Field Study investigated whether the participants in the CfD Study were able to use IntiVisor by themselves over an extended period of time. The study aimed to complement the CfD Study to further examine whether the objectives of the system and framework were reached.

5.3.1 Participants

The participants were the same as the ones in the CfD Study.

5.3.2 Data

All interaction data from the CfD visualization were available to the participants (Figure 48). By default, only the most recent two weeks of data were visualized. An analyst could load up to one month of data from any time period in the dataset using the data query feature in the control panel of IntiVisor (Figure 17c).

5.3.3 Process

I held three group study sessions in about one month apart remotely with the participants. A summary of the study sessions are listed in Table 10. The sessions each have a different set of activities for understanding the patterns and insights discovered in the CfD Study or for exploring new patterns in the Field Study.

Session 1

In this session, the participants and I revisited findings from the CfD Study. In these sessions, there were conflicting opinions from the participants, such as whether the sum-based aggregations should be used. Therefore, in this first Field Study session, I brought these findings to the group to have a discussion on whether any of these findings should inform a feature change in the CfD visualization. Furthermore, some features were introduced or reviewed in a set of training workshops for the users of the CfD visualization shortly after the CfD Study. The group examined how these features were adopted before and after the workshops to see how users reacted to the training sessions. This first session was conducted about two weeks after the training workshops so that we could examine up to two weeks of usage patterns after the training session.

Session 2

In this session, I conducted an exercise to explore higher-level analysis patterns with the group. I started by holding a discussion on what the participants envisioned a typical analysis session, good or bad, would be composed of. Next, I took a data-driven approach by directing the participants to review a few usage sessions using IntiVisor's Inspect view to help obtain ideas on which other activities might characterize a typical session. Last, I asked the participants to take a different approach by examining visual patterns emerging from a large group of sessions in IntiVisor's Explore view. By using these varying approaches, the participants discovered a broader set of usage patterns of varying levels of composition and abstraction from the CfD visualization usages that could be useful for informing future user trainings of the application.

Session 3

In this session, the participants and I explored a subset of patterns in IntiVisor that were identified from the previous (2nd) session. From examining the usage distributions of the patterns, the participants can gain a better understanding of the significance of each pattern within the dataset. Furthermore, I interviewed the participants about their experience using IntiVisor from the framework.

5.3.4 Patterns

Table 11 lists all the patterns that were discussed and/or explored by the participants using one month of data indicated at the top. In the second session of this study, I had a discussion with the participants about how they would characterize a typical analysis session. As a result, we were able to come up with lengthier, higher-level usage patterns. A subset of these "speculated" patterns that were of more interest to the participants, were explored in IntiVisor. The notations in the table are the same as in Table 7.

Table 8: Saved patterns with questionnaire responses of the CfD Study. (a) Numerical and color-coding in the questionnaire table. The questions were presented to users in this order in a UI dialog. The colors are chosen semantically to map to more interesting responses. (b) Questionnaire responses are organized in separate columns. Notice the Indication of Design Issue question was moved to the far right instead of in its original dialog position because it uses a different scale (0-2 instead of 1-3). Because the colors of responses are mapped semantically, an analyst can examine the spread of green to determine which patterns are more interesting to a participant. For example, the first pattern from the first participant, "Med sleep adding" was considered interesting in many aspects based on the spread of green in the responses. On the other hand, the red line at the right side of the table indicates that no design issues were found by any participant, which is a less interesting outcome.

1	Knowledge of existence	1	Low	2	Medium	3	High
2	Unexpectedness of usage amount	1	Low	2	Medium	3	High
3	Indication of design issue	2	Yes	1	Maybe	0	No
4	Insight significance	1	Low	2	Medium	3	High
5	Amount of effort	1	Low	2	Medium	3	High

(a)

Participant	Title	Level of Knowledge for Existence (1-3)		Level of Unexpectedness in Usage Amount (1-3)		Level of Insight (1-3)		Level of Effort (1-3)		Indication of Design Issue (0-2)	
1	Med sleep adding	Low	1	High	3	High	3	Medium	2	None	0
	Print	Medium	2	High	3	Medium	2	Low	1	None	0
2	save data?	Medium	2	High	3	High	3	High	3	None	0
	bookmarks	Medium	2	High	3	High	3	Medium	2	None	0
	week view	High	3	Low	1	Medium	2	Medium	2	None	0
	overnight	High	3	High	3	High	3	Medium	2	None	0
	removals	High	3	Low	1	Medium	2	Medium	2	None	0
3	add night shift	Low	1	Low	1	Medium	2	Medium	2	None	0
	total by month	Low	1	Medium	2	High	3	Medium	2	None	0
	line	High	3	Low	1	Medium	2	Medium	2	None	0
	intensity use	High	3	High	3	Medium	2	Low	1	None	0
	duration	High	3	High	3	Medium	2	Low	1	None	0
	week info	Medium	2	Medium	2	Medium	2	Low	1	None	0
4	removal	High	3	High	3	Medium	2	Medium	2	None	0
	trend	High	3	Medium	2	Medium	2	Medium	2	None	0
	sum by month	High	3	High	3	High	3	Medium	2	None	0
	annotation	High	3	High	3	High	3	Medium	2	None	0
	shift	High	3	Low	1	Medium	2	Medium	2	None	0
Average	-	2.44		2.28		2.39		1.83		0	

(b)

Table 9: Saved patterns of the CfD Study. Similar patterns are shown with the same color background and are placed in the same column. For example, similar patterns examining the night shift data are saved by three participants as shown as the three blue blocks in the second column.

Participant	Pattern						
1	Med sleep adding	Print					
2	overnight	bookmarks	removals	week view	save data?		
3	add night shift	total by month	duration	week info	line	intensity use	
4	shift	sum by month	removal	Drop down line	Drop down inten	annotation	trend

Table 10: Field Study sessions

Session	Number of CfD Participants	Activity
1	4	Discussed patterns saved in individual sessions of the CfD Study as a group. Examined features introduced and reviewed in the workshops conducted after the CfD Study.
2	3	Discussed patterns that characterize a typical usage session. Explored individual sessions to find such patterns.
3	3	Explored a subset of patterns discussed in the previous session in IntiVisor. Further analyzed new patterns that emerged in the session.

Table 11: List of patterns discussed in the Field Study. Patterns connected with "→" indicates that the interactions before and after the "→" need to occur back to back. On the contrary, patterns connected with "..." only needs the interactions before and after the "..." to occur in that order but not necessarily back to back. The amount of use by session and user of each pattern are listed in the corresponding columns. Interactions on the visualization views mostly occurred in the Timeline view unless they are specifically labeled with Circular view or Parallel view.

Field Study
Number of Users: 23
Number of Sessions: 356
7/16/16-8/16/16

Pattern	Length	Session	User
Sum data by month	1	69	12
Sum data by week	1	24	8
Add sleep	1	12	4
Toggle shift2	1	122	11
Average data by week	1	26	7
Add bookmark	1	6	4
Save bookmarks	1	5	4
Sum data by day	1	52	10
Add annotation	1	43	13
Print	1	176	20
Circular view (mouseover->mouseout line/bar)	2	0	0
Parallel view (mouseover->mouseout line)	2	2	2
Start-> Remove medication	2	5	3
Start-> Toggle shift2	2	17	6
Show shift2-> Hide shift2	2	24	6
Mouseover info-> Mouseout info	2	188	20
Connect webpage-> Drag and drop workbook-> Start visualization	3	67	13
Zoom in (y-axis)-> Remove medication-> Zoom time range	3	0	0
Start-> Remove behaviors-> Added night shift-> Average data by week-> Print	5	0	0
Show all data...No data (LOA, no data recorded, ?)	2	3	1
Add sleep...Zoom to time range	2	3	3
Configure to default (average data by month, behavior bars, medication to lines) -> Print	2	3	2
Toggle trend... Zoom to time range...Set trend (zoomed)	3	34	13
Start-> Show trend...Zoom to time range...Average data by month/week	4	0	0
Remove medication...Show trend...Set trend (zoomed)...Zoom to time range	4	16	8

Same as in the CfD Study, I generated two charts in Figure 55 to show the usage distributions of patterns over sessions and users. Figure 55a shows all the patterns over their amounts of users and sessions. Outlying patterns are labeled in the figure. Figure 55b shows a zoomed in view of Figure 55a with the labeled patterns in Figure 55a removed.

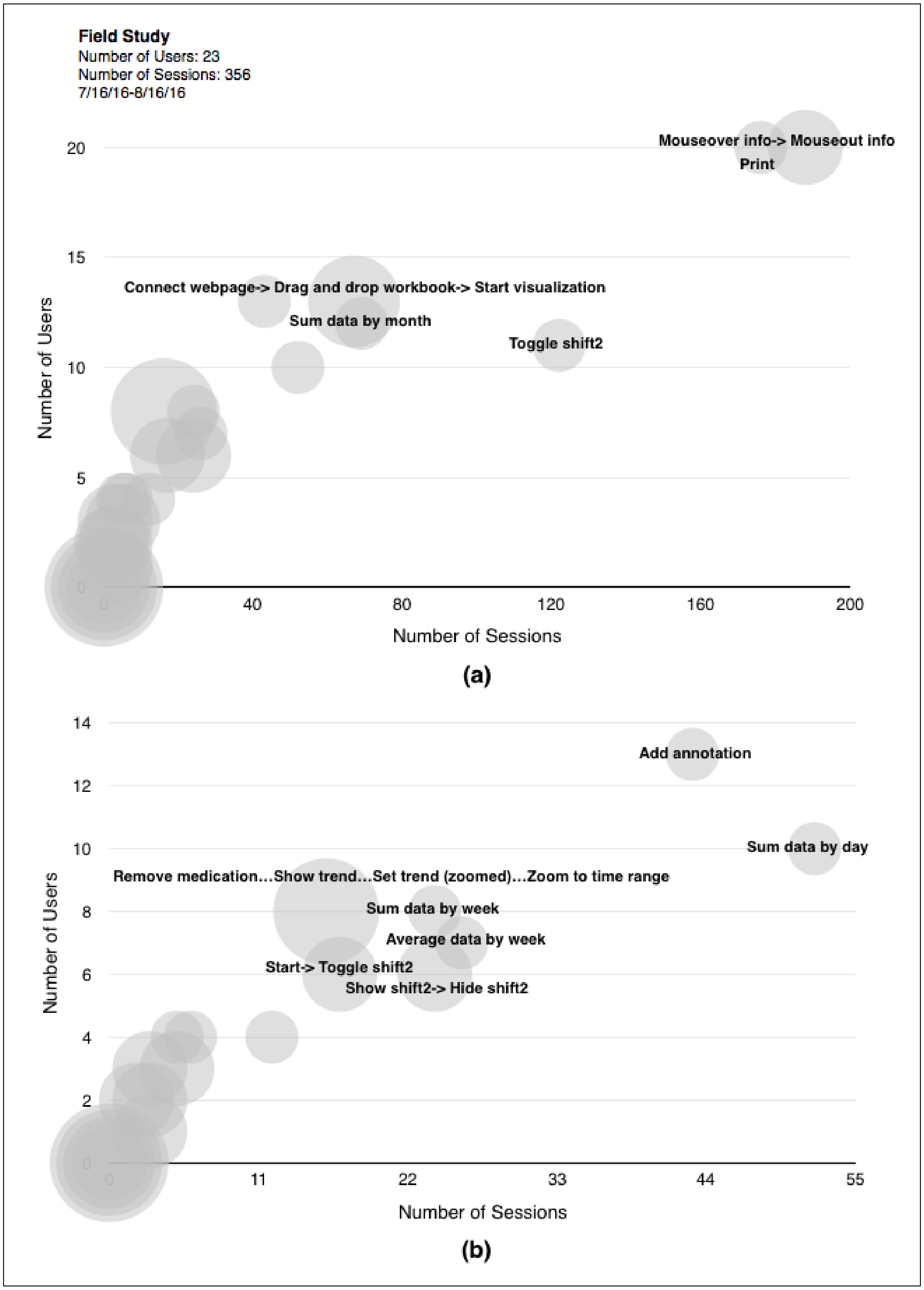


Figure 55: The amount of users and sessions of the explored Interaction patterns in the Field Study. The size of the circles are mapped to the length of the interaction sequence, disregarding whether the items in the sequence occurred back to back. (a) All patterns. (b) Subset of patterns without the labeled items in (a).

During the second session, we examined the effects of a set of training workshops. Figure 56 shows the session and user distributions of four patterns introduced or reviewed in the training workshops. These distribution charts were from the usage distribution view of IntiVisor (Figure 45) where the x-axis is time and the y-axis is the amount of sessions or users. The time window of the interaction data selected in the study session was about a month with the date of the workshops approximately in the middle. With this selection, we could explore the usage patterns before and after the workshops. Because these workshops were given in a room with computers where the CfD visualization users could try using the patterns, typically a peak in usages were present in the middle of the usage distribution.

5.3.5 Discussion

From the patterns discussed and explored in the Field Study, in this section I discuss how they answer the three research questions.

A. How do we provide flexibility in the analysis process for identifying patterns?

Using the same composition/abstraction dimensions in Figure 52, I examined the patterns identified in the Field Study in Figure 57 to demonstrate the flexibility of IntiVisor and the framework. From the figure, it is apparent that a higher portion of patterns discussed and/or explored in the Field Study had a higher-level of composition and abstraction. This distribution demonstrates that the participants were beginning to seek higher-level activities from the interaction data than in the CfD Study (Figure 53). Two reasons may have caused this change in pattern interests. First, the participants were encouraged to seek patterns that characterized entire sessions for training purposes. Second, the participants were becoming more familiar with the interaction data and analysis system to seek more complex patterns.

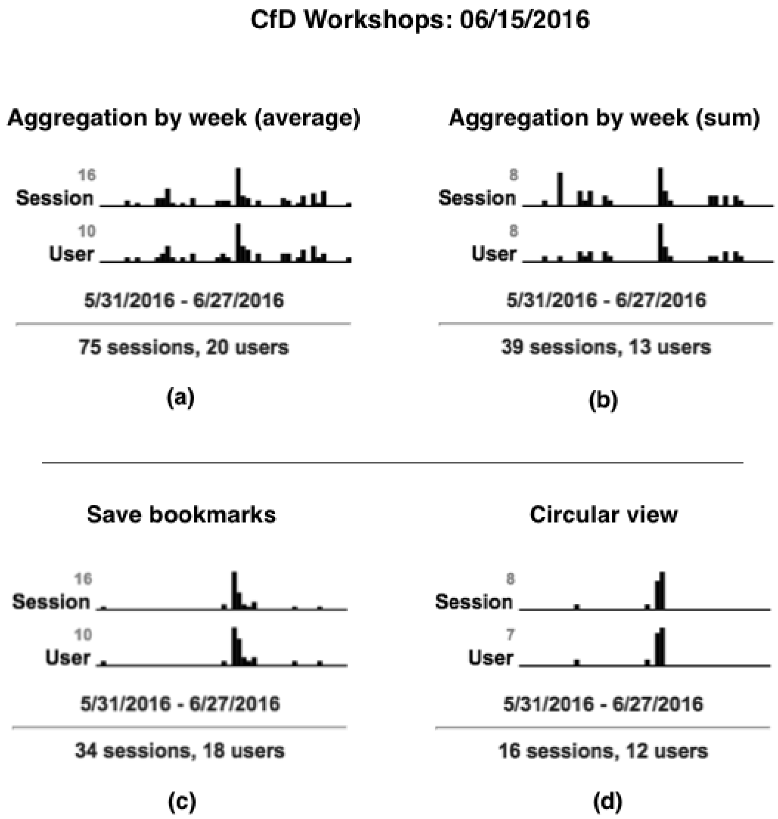


Figure 56: Session and user distribution before and after workshops. The workshops were given to the CfD visualization users on June 15th, 2016, approximately in the middle of the dataset selected (5/31/16-6/27/16). (a) Usage distributions of Aggregation by week (average). (b) Usage distributions of Aggregation by week (sum). (c) Usage distributions of Save bookmarks. (d) Usage distributions of Circular view (Mouseover → Mouseout bar/line).

Because the participants began to formulate more complex patterns, flexibility limitations of IntiVisor began to show. For example, one participant was seeking a spread of an issue where users incorrectly drew a trend line. In the CfD visualization, a trend line could be drawn with all the data loaded, which could be years of data, or drawn with a subset of the data within a specified time window. By default, the system drew trend lines with the entire set of loaded data. If a user zoomed into a specific time window and wished to redraw the trend line with only the visible data subset, the user needed to manually toggle a switch to accomplish that. However, in some cases, the users might have forgotten to toggle the switch and visualized a trend line that was incorrectly using the entire set of loaded data. What complicated matters was that this toggle could be applied any time before the zoom or the trend line display.

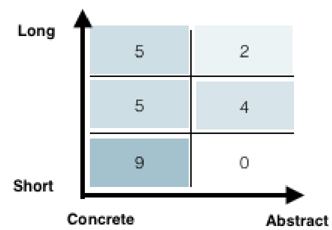


Figure 57: Organizing the patterns discovered from Field Study of the CfD visualization. The numbers in the chart areas show how many patterns are within the corresponding spaces.

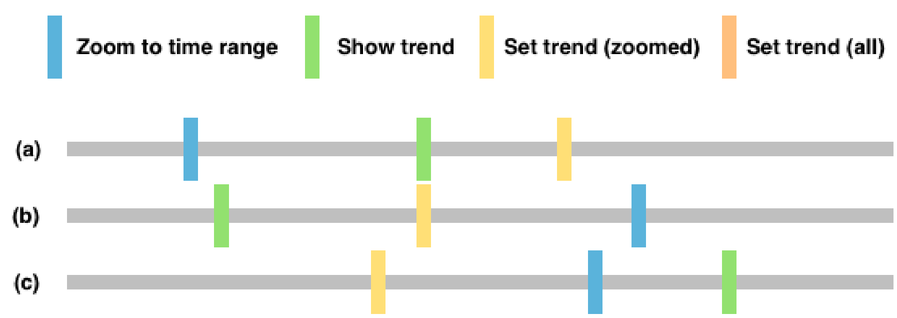


Figure 58: Interaction patterns for drawing trend lines using data in the visible view in the CfD visualization. The x-axis is time and the colored bars are interaction events. (a)(b)(c) show three different patterns. The last interaction event, Set trend (all), is the opposite of Trend (zoomed) should appear in the patterns for this use case.

To identify instances of this issue, an analyst needed to consider a set of patterns. In Figure 58, the following three patterns could all lead to the correct trend line being drawn. The design of the CfD visualization allowed the displaying and hiding of trend lines at any time. Two modes of a trend line were available. One was "zoomed," which drew trend lines based on the visible data within a "zoomed" time window. And the other was "all," which drew trend lines for all the loaded data, irrespective of the visible subset. This mode could be toggled at any time to indicate which type of trend lines should be displayed. Therefore, in Figure 58, at the end of the timeline, all patterns led to displaying trend lines of only the visible data. So to identify incorrectly drawn trend lines, an analyst sought patterns that did not include setting the trend line into the desired, "zoomed" mode before the end of the session. Unfortunately, IntiVisor did not have the ability to indicate an operation "NOT" being present in a pattern. As a result, the participants were not able to specify the specific pattern they were seeking. To work around the issue, the participants examined the presence of all the "Zoom to time range," "Show trend," and "Set trend (zoomed)" operations at the same time to visually identify the issue from the Explore view. The pattern could not be specified with the limited selection language in IntiVisor but was discoverable using the visualization.

B. Which types of insights can analysts gain from the identified patterns?

The insights found in this study are presented in three sections. In the first section, I present findings from analyzing the session and user distributions of the patterns. Next, I present group discussions of insights discovered in the CfD Study to examine collectively on whether these insights had any implications on the CfD visualization design. In the last section, I present findings from examining the effect of the training workshops for the CfD visualization users.

Usage Frequency Analysis

From the charts in Figure 55, several outlying patterns stood out. Many outlying patterns were already discussed in the CfD Study. For example, using sum as the aggregation method, examining the night shift data, and printing. But other patterns may be worth a closer look. For example, in Figure 55a, "Mouseover info→ Mouseout info" were the most frequently used feature, even more than printing! The pattern occurred whenever a user moved a mouse cursor over the "Info" label in the CfD visualization. This action brings up a view that displays an explanatory illustration of features in the view. At first glance, one might assume that the feature was very useful. But that did not make sense because the feature should not be frequently used. A more reasonable interpretation is that users frequently interacted with this feature by accident. The label "Info" was located right above the "Print" button. Therefore, users might have been accidentally interacting with the feature when reaching for the print button or other nearby features. This finding demonstrates the challenge in interaction analysis where an analyst needs to be careful with interpreting frequency-based observations.

One particularly interesting pattern discovered from IntiVisor was "Remove medication...Show trend...Set trend (zoomed)...Zoom to time range" in Figure 55. The participants found this pattern to be a characterization of a good analysis session that could be used in their trainings. The pattern showed that a user first removed one (or possible more) medications to clean up the view. Then, a trend line was drawn with data from within a selected time window, likely a recent time window at the time of the analysis. The pattern was used relatively extensively for a pattern of this length. Finding this valuable pattern and learning about its extensive usage was the benefit in using IntiVisor and the visual interaction analysis framework.

Implications for Design

In the CfD Study, many discoveries of individual patterns were considered insightful but none seemed to indicate a design issue to the participants (Table 8b). Furthermore, on several occasions, the participants did not seem to have the same interpretation on why certain patterns were used in the observed manner (Table 9). Therefore, in this study, we discussed some of these patterns to see if the group can collectively reach a better understanding of these patterns that were insightful to individual participants.

Several patterns that could have led to a design change eventually did not after the group discussions. For example, one participant found the extensive use of sums instead of averages to aggregate data reasonable but another participant was uncomfortable with it. As discussed in Section 5.2.5, the one participant that was fine with the finding was because the participant had a reasonable use case for it. Upon discussion, the other participant who was uncomfortable with the finding became less concerned knowing that there was a reasonable use of the pattern. This discussion enabled one participant to learn about the alternative use of the CfD visualization from another participant, which led to the decision that the design of the application was adequate. Another example was that many users seemed to have removed medications from the visualizations extensively. One participant provided an explanation of their removals (discontinuation). The point of discussion in the group was that then should the visualization application be redesigned to by default remove all medications that were discontinued? One participant disagreed because medications go full circles so that learning about which medications were discontinued was still useful. As a result, the final decision was to keep the design and let users manually remove discontinued medications if and when they wish to do so on a case by case basis.

On the other hand, one pattern led to a design change in this study. The trend line example illustrated in Figure 58 showed that users might have been forgetting about a toggle in the CfD application to draw trend lines with only the visible data. Upon discussion and examination of its impact using IntiVisor, the participants agreed that it made sense to change the default configuration to draw trend lines using the visible data instead of all the data. After the study session, the CfD visualization was updated accordingly.

Training Workshop Effect

I gave a set of training workshops to the CfD visualization users right after the CfD Study. These workshops were designed to introduce new features and to review less-used features. Using IntiVisor, the participants examined the usage distributions of the features presented in the workshops to assess the impact of the training sessions. The room where the workshops were held had desktop computers for users to try using the features on their own. During the first Field Study session, we examined about one month of data that included interactions from two weeks before to two weeks after the workshops (Figure 56).

One new feature that was expected to be widely used was data aggregation by week. By default, the CfD visualization aggregates (sum/average) data by month and supports interactively changing this aggregation level to by day or by shift (three per day). Unfortunately, sometimes the aggregation level was either too high (month) or too low (day). As a result, a CfD visualization user requested the week-based aggregations so it was expected to be very useful. The feature was available for about two weeks ahead of the workshops but was never formally introduced to the users until the workshops. Therefore, the expected usage distribution was it being used little before the workshops, tested extensively during the workshops (using computers in the room), and then more broadly used after the workshops. In Figure 56a and Figure 56b, the session and user distributions from the two types of data aggregation by week (sum/average) are shown.

Disappointingly, the usage distribution of the week-based aggregations did not seem to match the expectation well. Although the peak usage in the middle matched the expectation of an extensive test period during the workshops, the amounts of sessions and users seem visually comparable before and after the workshops. One hypothesis for this observation was that the users who wished to use this feature already somehow obtained the knowledge of its availability informally before the workshops so that they were already using it then. As a result, after the workshops the same subset of users continued to use it the same way. But from this view, it was not possible to know whether the users after the workshops were the same set of users before the workshops. To verify this hypothesis, I examined the users in the Explore view of IntiVisor before and after the workshops. It turned out that for both patterns, three new users started using the feature and two earlier users did not use it after the workshops. This observation indicated that even though the workshops did not seem to increase overall usage, more users were actually picking up the feature as expected. Furthermore, a subtle but significant difference between Figure 56a and Figure 56b was that there seemed to be nearly twice as many sessions (75 vs. 39) and users (20 vs. 13) that used averages than sums! This difference can be gleaned from the y-axes scales of the charts or the labels in the bottom. The reason why averages were used more than sums may be attributed to the fact that averages considered missing data as discussed in Section 5.2.5.

Two of the patterns that were introduced in the past were reviewed in the workshop due to their lack of use from the CfD visualization users (Figure 56c and Figure 56d). One pattern reviewed in the workshops was for saving bookmarks. As shown in Figure 56c, some users attempted to use it soon after the workshops, perhaps to gain more knowledge about the feature by themselves. In the charts, it might seem that no new users picked up the feature. But upon further examination of the users before and after the workshops (1 at the left and 2 at the right) in the Explore view of IntiVisor , those users were actually different. This observation indicated that the feature was being adopted by more users, showing the success of the workshop in helping users pick up the feature. On the other hand, the Circular view was unfortunately completely refused by the users (Figure 56c, Circular view: mouseover → mouseout bar/line). The peak usage showing users testing the feature in the workshops was apparent but no usages were observed afterwards. The Circular view used an advanced visualization design that could take a while for users to interpret the patterns as well as to communicate findings to others (Appendix A.2.4). As a result, despite an extensive effort in improving and educating users about the view, the usage data showed that it was not as useful as expected.

C. How do we provide analysts the capability to practically identify and analyze patterns?

I found that only one participant spontaneously used IntiVisor once. In this 10-15 min session, the participant was able to iteratively use all the views, most of them multiple times (Inspect: 1, Select: 2, Categorize: 4, Sequence: 2, Explore: 5), to identify patterns of interest. Because I wished to examine whether participants were able to spontaneously use IntiVisor, I did not ask them to use it at least a certain amount of times as part of the study. The result showed that the system was apparently not useful or not easy to use enough for most of the participants to even try it on their own. The reasons why the participants might have found IntiVisor too challenging to learn and adopt might be because they did not usually analyze interaction logs and operate such expert visual analytics systems. The interactions available in the CfD visualization were moderately complex over two years of development. Although the limited adoption of the system was disappointing, it was encouraging to see at least one participant using the system in an iterative manner.

In the last session, I interviewed the participants to learn about their experiences. Even though most of the participants were not using IntiVisor on their own, they still participated in the group sessions where we discussed and explored patterns together. I mostly operated the interactive features in these sessions to not let the learning curve of the system impact the analysis. The one participant who used the system outside of the group sessions was comfortable using the basic features of the system and thought it already provided good information. But another participant emphasized the need of assistance on using the system and suggested including explanatory information in the user interface to help with the learning process. These interviews highlighted the challenging aspect on learning the features of IntiVisor.

5.3.6 Conclusion

The Field Study did not have enough independent usages of IntiVisor to explore how participants were the system by themselves. However, I was able to further explore the flexibility of the system and framework in identifying more complex, higher-level patterns in the group sessions. The participants were able to discuss patterns and insights discovered in the individual sessions in the CfD Study as well as collectively explore how certain features were adopted after a set of training workshops. Because IntiVisor was a technical visual analytics system designed to support an unfamiliar task (interaction analysis) with an interaction dataset of a moderately complex visualization application (CfD visualization), the CfD participants did not seem to be able to adopt it. As a result, in the Generalization Study, I applied the framework to other, simpler visualization applications with participants that might be more comfortable with using expert visual analytics systems.

5.4 Generalization Study

The Generalization Study investigates whether the visual interaction analysis framework can be effectively applied to other visualization applications. This study also includes a set of users with computer science backgrounds who developed the applications so I could further investigate whether IntiVisor was practical enough for this type of users.

5.4.1 Visualization Applications

The Generalization Study includes four visualization applications. CiteVis[5] visualizes the citation patterns of visualization publications using a grid layout of circles.

[5] CiteVis: http://www.cc.gatech.edu/gvu/ii/citevis/

VISLists[6] visualizes a broader set of visualization publications but using a list-based layout, similar to that in the List view of Jigsaw[44]. List View[7] uses the same visualization technique as VISLists but allows users to import their own datasets. Microsoft Research Data Visualization Apps for Office (Office Visualizations) include a set of visualizations that can be used in Microsoft Excel and Microsoft Access[8]. The apps include a long list of visualization techniques: histogram, 2D-histogram, treemap, streamgraph, line chart, area chart, bar chart, column chart, scatter chart, and pie chart. These visualization applications may not have as many available interaction types and usages as the CfD visualization but they provide other types of visualization techniques and different sets of interactions.

5.4.2 Participants

The participants of this study are the designers of the corresponding visualization applications. These applications were deployed online so before the study, the participants had little to no knowledge about how their applications were being used remotely. One designer of CiteVis, two designers of VISLists and List View, and one designer of the Office Visualizations participated in the study (Table 12). Because the participants of VISLists and List View were the same. I was able to study how the participants would analyze the two applications differently.

5.4.3 Data

For CiteVis, VISLists, and List View, I worked with the participants to instrument their visualization application to log their interaction data. For the Office Visualizations, the participant already had log data. The amount of interaction data collected in each visualization application for the studies and participants are illustrated in

[6]VISLists: http://www.iilabgt.org/vislists
[7]List View: http://www.iilabgt.org/listview
[8]Microsoft Research Data Visualization Apps for Office: https://www.microsoft.com/en-us/research/project/microsoft-research-data-visualization-apps-for-office/

Table 12: Visualization application and interaction data used in the study for each participant.

Participant	Visualization	Interaction Data
1	VISLists	~ 2.5 months
	List View	~ 2 months
2	VISLists	~ 3.5 months
	List View	~ 3 months
3	CiteVis	~ 4.5 months
4	Office Visualizations	~3.5 years

Table 12. As shown in the table, there were at least multiple months of interaction data logged for each visualization application. For CiteVis, VISLists, and List View, one participant of each application worked with me to log interactions in their applications. Because these three visualization applications were freely available online without the need of a login, the participants could not explicitly log a user identification as in the CfD visualization to differentiate the users. For the Office Visualizations, the participant was interested in learning about "the story of charts." As a result, we used a unique identifier of each chart as users in the system for the analysis.

5.4.4 Process

This study was conducted the same way as the CfD Study. I met each participant individually in a two-hour session for each visualization application (Table 3). I first walked through the features of IntiVisor and then let the participants operate the system by themselves while thinking aloud about their analysis processes.

When specific patterns were of interest to participants, I asked them to save it and answer a short five-question questionnaire (Table 5). An interview was conducted at the end of the study to learn more about the participants impressions about the framework. Additionally, the states of IntiVisor were logged whenever a participant changed views or manually saved a session in the system.

5.4.5 Patterns

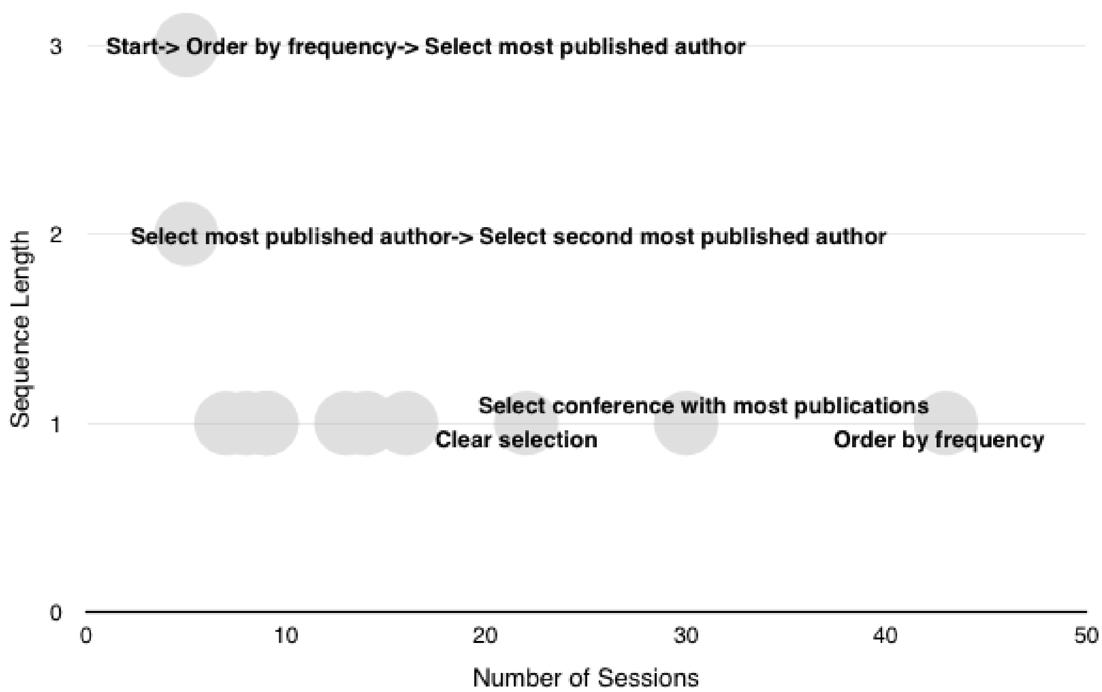
For each visualization application, I collected patterns explored in Figure 59a (VISLists), Figure 60a (List View), Figure 13 (CiteVis), and Figure 62a (Office Visualizations). Each table lists the patterns and the corresponding amount of sessions that include the patterns. The total number of sessions and time frame of the data analyzed in these tables are listed at the top. For example, for VISLists (Figure 59a), 203 sessions were included in the analysis. The notations in the table are the same as in Table 7. Note that the usage frequencies are extracted to exemplify the relative amount of usages of the patterns explored so they are not the same as what the participants examined at the time of their study.

For CiteVis, VISLists, and List View, I distributed patterns over their lengths and the number of sessions that included them, as shown in Figure 59b (VISLists), Figure 60b (List View), and Figure 61 (CiteVis). These charts are different from those used for analyzing the patterns in the CfD Study (Figure 51) and the Field Study (Figure 55) because I cannot distribute the patterns over users without the user identities explicitly logged. For the Office Visualizations, I distributed patterns over the number of charts and sessions that included them (Figure 62b).

VISLists
Session Number: 203
2/19/16-8/30/16

	Length	Session
Select multiple items	1	8
Clear selection	1	22
Select conference with most publications	1	30
Select most published author	1	14
Select second most published author	1	7
Order by hasStrength	1	13
Change text alignment	1	9
Change dropdown list	1	16
Click align text	1	9
Order by frequency	1	43
Select most published author-> Select second most published author	2	5
Start-> Order by frequency-> Select most published author	3	5

(a)



(b)

Figure 59: Patterns explored in interaction data from VISLists. (a) Table of patterns with corresponding amounts of users and sessions. (b) Charting patterns over the number of sessions that include them over their lengths.

List View
Session Number: 388
3/9/16-8/25/16

	Length	Session
Change font	1	8
Change text alignment	1	17
Clear selection	1	29
Upload	1	35
Start-> Upload	2	34
Start-> Sample dataset 1	2	32
Start-> Sample dataset 2	2	41
Start-> Sample dataset 3	2	21
Change dropdown list-> Order by frequency	2	15
Change dropdown list-> Order by name	2	3
Upload-> Upload	2	7
Connect-> Connect	2	137
?...Upload (end)	2	14
Select item-> Order by frequency-> Select item...Select item	4	10

(a)

(b)

Figure 60: Patterns explored in interaction data from List View. (a) Table of patterns with corresponding amounts of users and sessions. (b) Charting patterns over the number of sessions that include them over their lengths. Note that “Connect” is the same as “Start.” I named it “Connect” in the “Connect→ “Connect” pattern because it makes more sense than “Start→ Start.”

Table 13: Patterns explored in interaction data from CiteVis. (a) Table of patterns with corresponding amounts of users and sessions. (b) Charting patterns over the number of sessions that include them over their lengths.

CiteVis
Session Number: 197
5/2/16-8/27/16

	Length	Session
Activate paper	1	69
Deactivate paper	1	22
Dropdown concept	1	31
Dropdown author	1	33
Dropdown affiliation	1	33
Dropdown affiliation (other than empty)	1	33
Dropdown affiliation ()	1	5
Dropdown concept ()	1	5
Dropdown author ()	1	6
Clear (Button, Shortcut)	1	25
Mouseover paper-> Activate paper	2	68
Dropdown author-> Mouseover paper	2	42
Mouseover paper-> Dropdown author	2	28
Activate paper-> Search keyword	2	0
Mouseover paper-> Activate paper-> Deactivate paper	3	7
Dropdown author…Activate paper	2	19
Dropdown author…Mouseover paper…Activate paper	3	19
Dropdown author…Activate paper…Clear (Button, Shortcut, Deactivate paper)	3	8
Dropdown author…Activate paper…Clear (Button, Shortcut)	3	4
Dropdown affiliation…Activate paper…Clear (Button, Shortcut)	3	3
Dropdown concepts…Activate paper…Clear (Button, Shortcut)	3	3
Dropdown author…Clear (Button, Shortcut)…Dropdown author	3	6
Dropdown affiliation…Clear (Button, Shortcut)…Dropdown affiliation	3	5
Dropdown concept…Clear (Button, Shortcut)…Dropdown concept	3	3

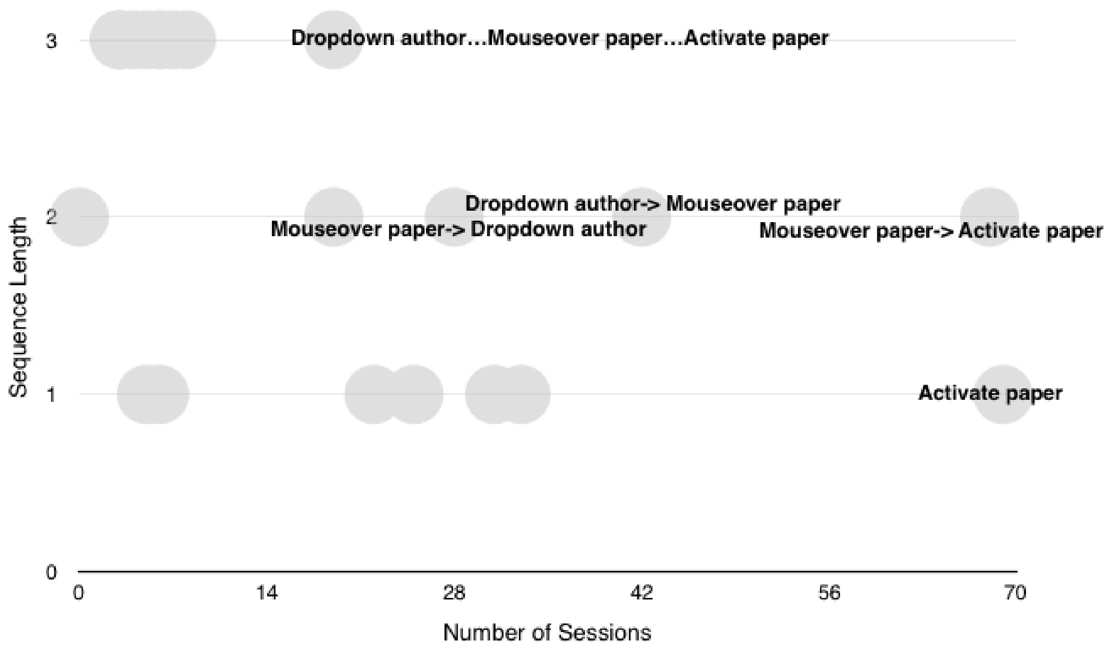


Figure 61: Patterns explored in interaction data from CiteVis charted over the number of sessions that include them over their lengths.

Office Visualizations
Session Number: 5356
7/1/16-7/31/16

	Length	Session	Chart
Generate failed tree (treeStats (rootCount 0))	1	871	788
Generate boring tree (treeStats (maxDepth 1))	1	355	314
Generate interesting tree (treeStats)	1	400	336
Generate sample tree (treeStats (rootCount 7, leaveCount 31))	1	1	1
Generate sample data (writeSampleData, writeSampleData2)	1	409	403
appInit (treemap)-> create	2	645	615
Error (error, warning)-> Generate interesting tree	2	62	55
Generate failed tree-> bindFromPrompt	2	207	207
appInit (treemap)-> Generate sample data-> bindFromPrompt	3	0	0
Generate interesting tree-> readBoundData (nodes, color, size)-> Error	3	35	35
readBoundData (color)-> readBoundData (nodes)-> readBoundData (color)	3	118	95
AppInit-> Create-> Manipulate (tooltip, resize, slider)-> Generate failed tree	4	18	18
load...bindFromPrompt	2	16	12
bindFromPrompt...load	2	213	209
Generate sample data...Generate sample tree	2	1	1
Generate failed tree...Generate sample data...Generate sample tree	3	1	1
load...readBoundData (nodes, color, size)...readBoundData (nodes, color, size)... readBoundData (nodes, color, size)...readBoundData (nodes, color, size)	5	142	97

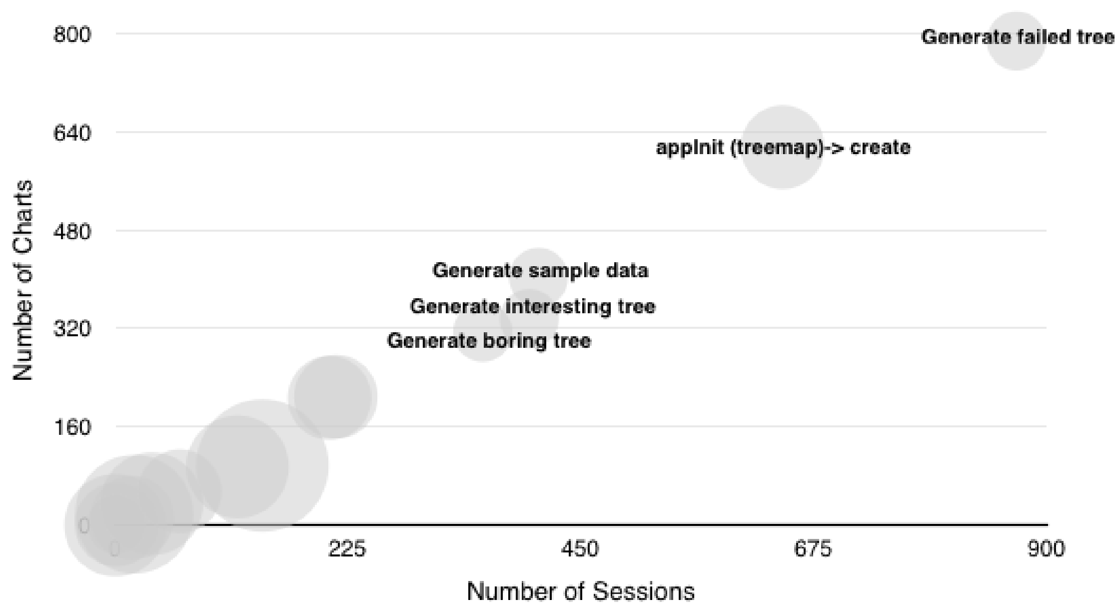


Figure 62: Patterns explored in interaction data from the Office Visualizations listed in a table (a) and charted over the number of sessions and charts (b).

Same as in the CfD Study, a subset of explored patterns were saved by participants. The same set of questions needed to be answered whenever a pattern was saved (Table 5). Table 14b lists the saved patterns and questionnaire responses. The responses are numbered and color-coded with a traffic light metaphor using Table 14a. I collected responses from one participant for each visualization application.

5.4.6 Discussion

I analyzed the explored patterns and combined the analysis results with the participants' impressions. These findings are organized by how they can answer the research questions.

A. How do we provide flexibility in the analysis process for identifying patterns?

To demonstrate the level of flexibility in IntiVisor, I present the variation of patterns discovered and perspectives created by the participants of the visualization applications.

Variation of Patterns

Same as in the CfD Study, I organized patterns explored into the composition/abstraction chart (Figure 52) in Figure 63. If IntiVisor was able to flexibility support the analysis process, the explored patterns should be more evenly distributed into the different areas in the chart.

- VISLists

  From Figure 63a, the patterns explored in VISList were heavily focused on (individual) events with little composition and abstraction. I suspect the lack of highly composed and abstracted patterns may be because the data values of selected items (e.g., specific authors) took the focus away from the more complex feature-usage patterns. VISLists uses a visualization publication dataset that includes information about the data values, such as the most published author, of selected items. Therefore, the participants focused their analyses on individual events that include those values. This preference was evident in the perspectives they generated in Figure 64, which will be discussed in the next section.

- List View

  From Figure 63b, the patterns explored in List View were more composed. List View is an advanced version of VISLists where users can upload their own data or select from a set of sample datasets. Because the participants chose not to log any data uploaded by their users, they could not tell the value of the list items selected as in VISLists. As a result, the focus of the analysis shifted to the features of the application, which increased the exploration of more complex patterns that included more than one feature. For example, "Change dropdown list→ Order by frequency" in Figure 60a shows a pattern where a user orders a list by the occurrence frequency of the list items after setting the variable (e.g., author) of that list. This finding is encouraging because it demonstrated that IntiVisor was able to help the participants identify a set of patterns with a higher level of composition with this visualization application.

- CiteVis

  From Figure 63c, the patterns explored in CiteVis were even more diverse in their composition and abstraction. This diversity could be because CiteVis can be considered more complex than VISLists and List View in some aspects. For example, CiteVis supported multiple ways of clearing selections in the application, such as clicking a "Clear All" button, pressing a keyboard shortcut ("c"), or deselecting (deactivating) a selected item. Therefore, if an analyst only cared about whether a clearance was performed instead of how it was performed, he/she could merge these varying ways of selection clearance into an abstract category, "Clear." This type of merging was used by the participant. Furthermore, the participant was particularly interested in patterns involving the combination of selecting authors, affiliations, and concepts with a clearance of selections. The extensive explorations of these varying patterns showed how IntiVisor was able flexibly support the analysis of yet another type of visualization application.

- Office Visualizations

  From Figure 63d, the patterns explored in the Office Visualizations were also quite diverse in their levels of composition and abstraction. Notably, the participant spent a significant amount of time exploring frequent sequences of consecutive operations in the Sequence view so that a large portion the explored patterns include more than one operation. Furthermore, the participant took full advantage of the categorization feature that supports the abstraction of interactions. For example, the participant categorized three actions, "tooltip," "resize," and "slider" into a category and labeled them "manipulate" on the canvas in the Categorize view (Figure 67). From the explored patterns, it was clear that this participant was able to utilize the categorization and sequence analysis features to effectively identify patterns of varying levels of composition and abstraction, demonstrating again the flexibility of the system and framework.

Variation of Perspectives

Another demonstration of flexibility in IntiVisor is to support a variety of perspectives. Figure 64, Figure 65, Figure 66, and Figure 67 show the variety of perspectives created

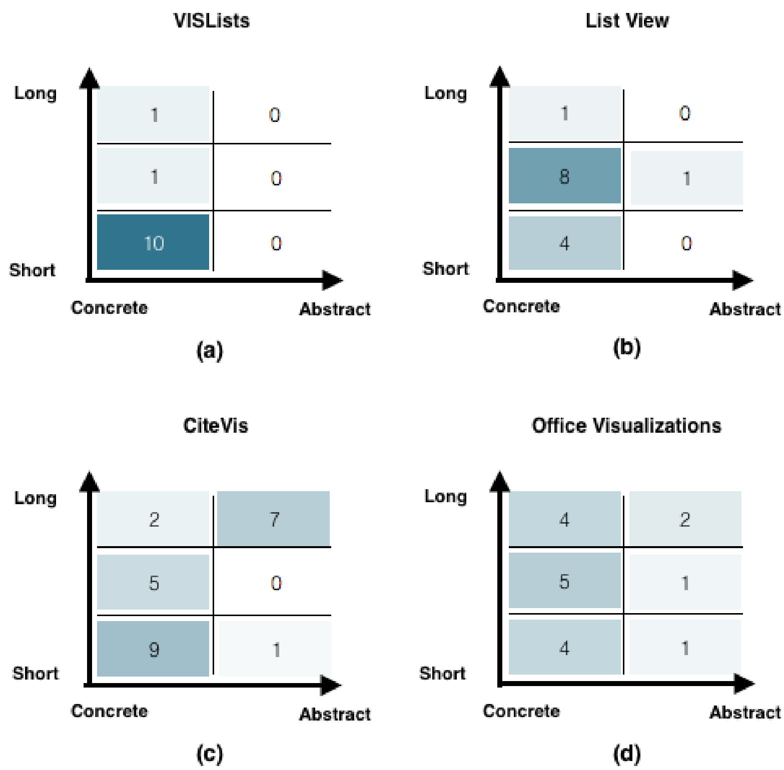
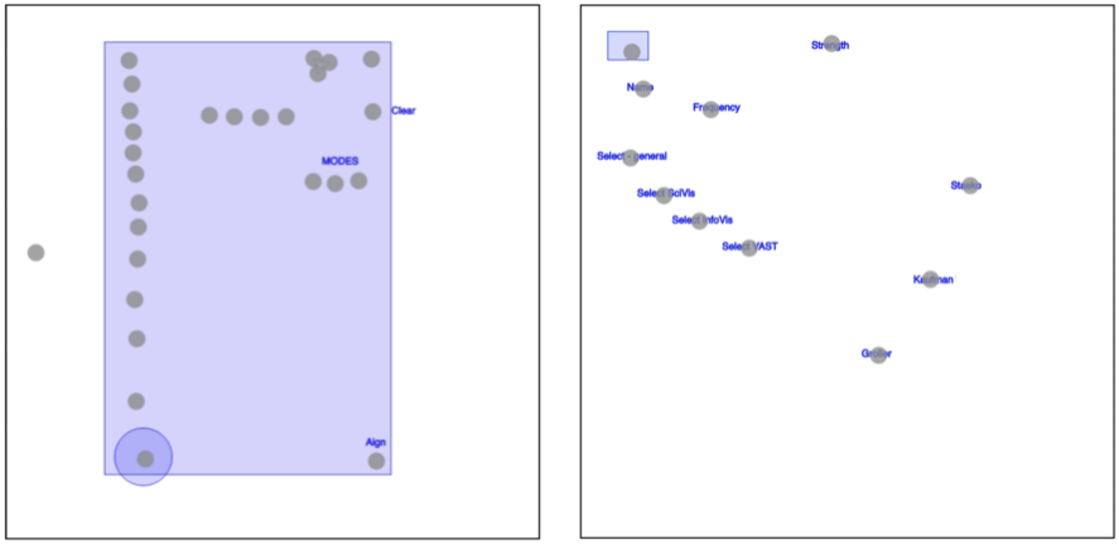


Figure 63: Patterns discovered from the Generalization Study mapped to the spaces in the composition/abstraction chart. The numbers in the chart areas show how many patterns are within the corresponding spaces in each visualization application. The background shades of the chart areas show the relative pattern amounts in a area compared to the other areas within the chart. (a) VISLists. (b) List View. (c) CiteVis. (d) Office Visualizations.

in IntiVisor in the Generalization Study.

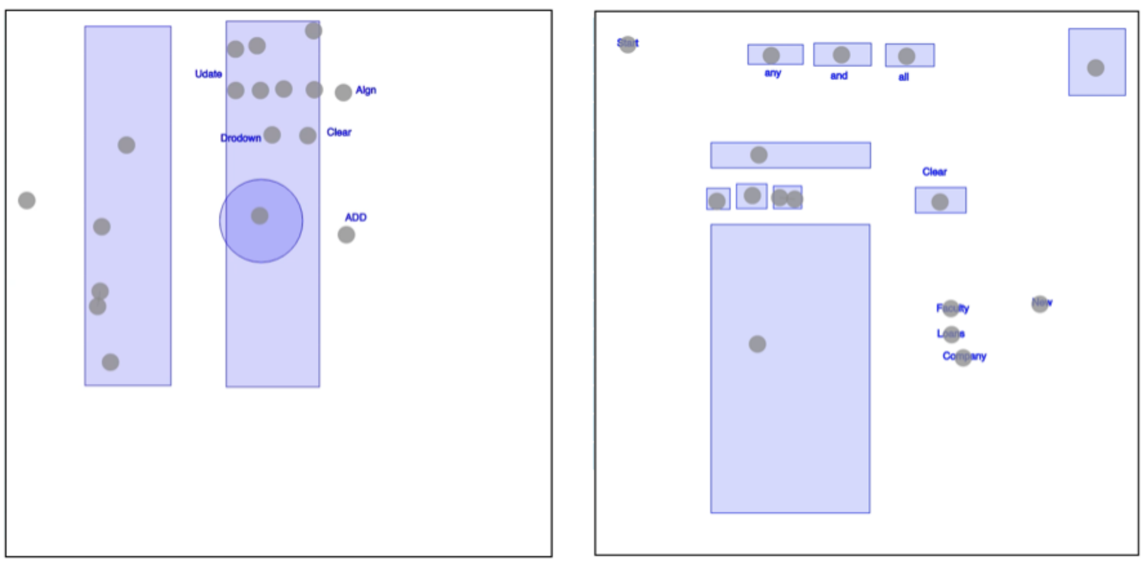
- VISLists

  The two participants that designed and analyzed VISLists' interaction data generated very different analysis perspectives (Figure 64). For example, participant 1 created a long list of operations that mostly indicate specific values (e.g., most published author) of selected items on the left side of a giant rectangle (Figure 64a). No labels were drawn for these operations. The start of the session was indicated by the lone circle to the left, and other UI-related features were organized in the space on the right side of the rectangle. For example, the top row was for changing dropdown list selections (e.g., conference → author) and



(a) (b)

Figure 64: Perspectives generated from the participants analyzing interaction data from VISLists. (a) Perspective by participant 1. (b) Perspective by participant 2.



(a) (b)

Figure 65: Perspectives generated from the participants analyzing interaction data from List View. (a) Perspective by participant 1. (b) Perspective by participant 2.

the second row was for the different ways of ordering list items (e.g., order by frequency). On the other hand, participant 2 listed a number of selected items but specifically labeled the values of the items on the canvas as context (e.g.,

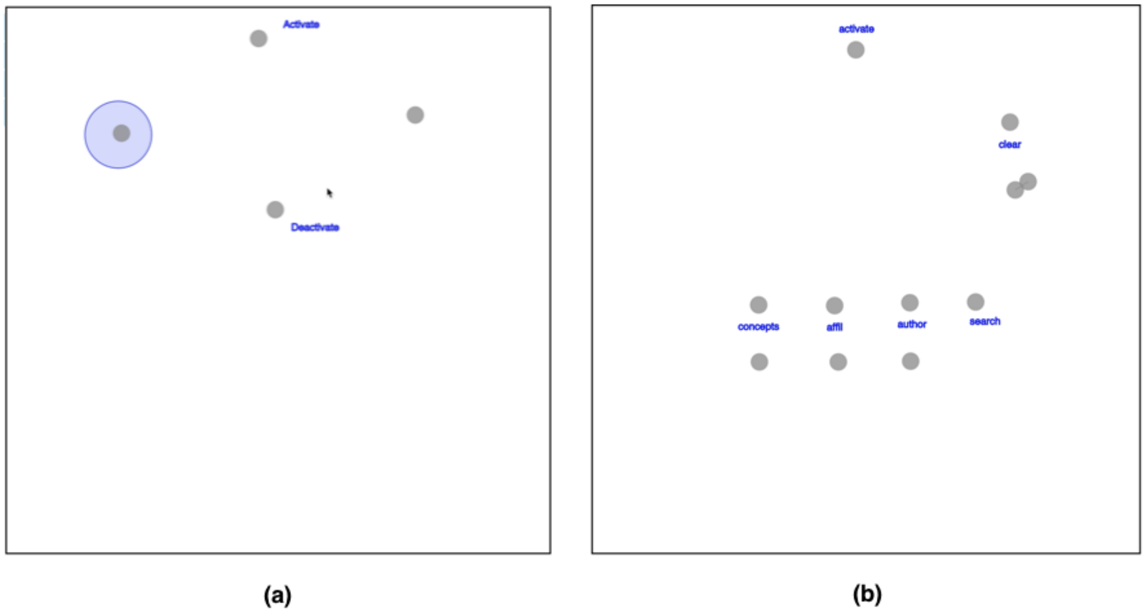


Figure 66: Perspectives generated from the same participant analyzing interaction data from CiteVis. (a) First perspective. (b) Second perspective.

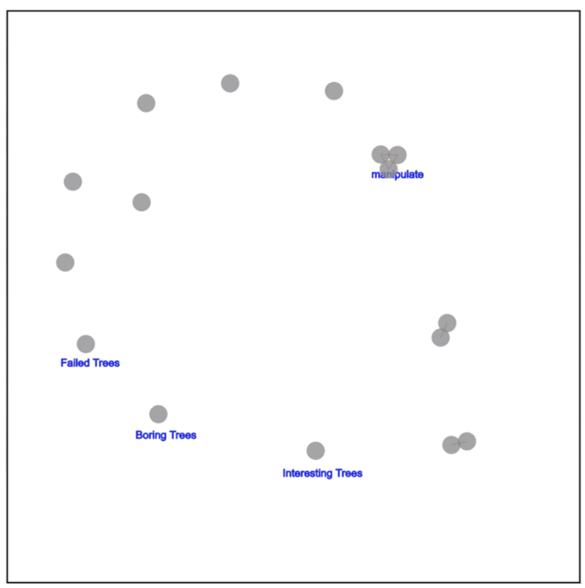


Figure 67: Perspective generated from the participants analyzing interaction data from the Office Visualizations.

Select InfoVis), as shown in Figure 64b. What was particularly intriguing was that this participant arranged these items in a circular layout because the participant knew that in a later part of the system, these items may be connected with tapered edges to show frequent sequences. A circular layout can minimize the overlapping of edges when these sequences are drawn. Then, three types of list ordering methods (Name, Frequency, Strength) were labeled at the top of the view while the start of the visualization was placed on the upper-left corner of the canvas. Comparing the canvases of these two participants, it was clear that the two participants took different approaches in labeling their operations. Participant 1, who instrumented the visualization application to log interaction events, had a general interest in the features used. On the other hand, participant 2 selected a very specific set of interactions—ordering methods and selected items—and ignored all the other interactions. These two perspectives showed that the canvas was able to support the participants' varying interests in the interaction data and how the data should be organized spatially.

- List View

  In contrast to VISLists, the same two participants of List View generated similar analysis perspectives (Figure 65). As opposed to VISLists, the interaction data logged in List View did not include data values. Without data values, the participants turned their attentions to the features of List View. They both sketched a context that approximately mirrored the UI layout of List View and mapped the corresponding interactions onto the layout. Of particular interest were the new features in List View—upload and select sample datasets—occurred on a separate page before the main visualization was loaded. As a result, the participants creatively mapped them onto the canvas in different ways. Participant 1 used the left rectangle, which was probably originally representing a list in the List View, as a dedicated space to spread out these data import options (5 circles, Figure 64a). On the other hand, participant 2 used an empty space to the right to map to these options: load sample datasets (e.g., faculty) and upload "new" datasets. The similarity of using the feature layout and dissimilarity in mapping the data import options demonstrated further how the canvas was able to help participants flexibly create the perspectives they needed for their analysis. It was surprising to see such different approaches in creating the perspectives in List View from that in VISLists by both the participants.

- CiteVis

  Different from the VISLists and List View participants, the participant analyzing CiteVis created two perspectives (Figure 66). The first perspective was generated using four events of interest to the participant (Figure 66a). The focus was on the inspection (mouseover, mouseout) and selection (activation, deactivation) of papers. Afterwards, a change in analysis prompted the participant to recreate the perspective from scratch. The new perspective included a different set of operations with a strong interest in the highlighting of papers with certain concepts, affiliations, or authors (Figure 66b). For each highlight type, there was a corresponding "unhighlight" event that was placed right below the label. Although the participant used minimal drawings on the canvas, the semantic meanings of the operations were sufficiently clear with their strategic placements. IntiVisor was designed specifically to allow analysts to dynamically change their analysis perspectives based on their analysis goals. Seeing that the participant was able to change the perspective on the fly showed how the system was able flexibly support the dynamic interaction analysis process.

- Office Visualizations

  The participant analyzing the Office Visualizations organized operations in a circular layout, as shown in Figure 67. A circular layout allows each operation (node) to have a direct path to connect to other operations. The analysis was focused on treemap usages. Therefore, several types of trees were singled out. For example, a "Failed Tree" had 0 root nodes, indicating a tree failed to be created and a "Boring Tree" had a depth of 1. The tree generation operations were positioned and labeled at the bottom of the perspective. Other actions were positioned around the circular layout. Some events, such as error and warning messages, are categorized into one operation (middle-right unlabeled two circles). The participant used a few labels but did not draw any geometric shape in the perspective. The choice of laying out events in a circular layout was similar to that of one of the VISLists participants (Figure 64b) and demonstrated again the flexibility of the canvas in supporting the analysis of yet another visualization application. The participant did question, however, whether the level of customization in this view was necessary because a circular layout seemed to be sufficient to the participant that could be automatically configured.

Limitations

From the study, participants brought up several limitations as opportunities to improve IntiVisor. For example, the participants who analyzed CiteVis and Office Visualizations data wished to include a duration-based criterion, such as a minimal of two seconds between events, to filter patterns. This type of criterion can be useful, for example, for differentiating "accidental" and "intentional" behaviors. Take CiteVis for example, whenever a user moves the mouse cursor over a paper, other papers that cited or was cited by this paper are highlighted. The highlight is removed when the mouse cursor is moved out of the paper. This interaction sequence is so simple that a user can accidentally invoke a long list of such sequences by simply moving the mouse cursor across the visualization. One key characteristic of such accidentally invoked sequences is that the duration between the two events are typically very short. If an analyst does not care about this type of accidental events, these events should be filtered out with a duration-based criterion. Unfortunately, the system did not support such filters. Another desired feature that was not supported by IntiVisor was the explicit grouping of sessions by pattern. When a group of sessions containing a certain pattern can be grouped and separated from the other sessions, they could be separately analyzed and compared to the other sessions. The process allows the analysis of new patterns under the condition that the grouping pattern existed, or did not exist, in those sessions. The participant analyzing interactions of the Office Visualizations hoped the system can support this type of "cohort analysis." These limitations could be addressed by new features in future work to improve the level of flexibility of the system.

B. Which types of insights can analysts gain from the identified patterns?

I present insights found in this study in two sections. In the first section, I present findings from usage distributions of explored patterns in each visualization application. In the second section, I present findings based on the questionnaire responses for the saved patterns.

Usage Frequency Analysis

For VISLists, List View, and CiteVis, I charted explored patterns over their lengths and the number of sessions that included the patterns. The frequency distribution over the length of a pattern could reveal much about whether a certain pattern is used proportionally to its complexity. Generally speaking, if a lengthy, highly composed pattern is used extensively, it might be a pattern that characterizes a pervasive visual analysis method used in the visualization application. For the Office Visualizations, explored patterns were charted over the number of charts and sessions that included the patterns. These two variables should be positively correlated so if any pattern deviates from the expected relation, it might reveal an unexpected finding.

- VISLists

  The patterns explored in VISLists were charted over sessions and lengths in Figure 59b. A key observation is that identifying "mosts" seemed to be a prevalent activity. This information can be gleaned from the extensive use of "Order by frequency" and the frequent selection of a conference with the most publications. Among the patterns explored, the "Sort by frequency" activity was the most frequently used pattern, in about 21% of the sessions (43/203). This percentage may not seem high, but if we remove about 60% of sessions (121/203) that have ≤ 3 events, the portion of "Sort by frequency" becomes about 44% (36/82)! This observation raised the question on whether sorting the lists by frequency should be made the default configuration. Furthermore, of particular interest to the participants was the selection of sequences that include the most published authors. But very few sessions were found that include these two patterns (2.5%, 5/203 for each pattern).

- List View

  The patterns explored in List View were charted over sessions and lengths in Figure 60b. List View had nearly twice as many sessions as VISLists (388 vs. 203) within a shorter deployment time window ($\sim$3 weeks). One clear outlier the participant was interested in was the appearance of consecutive connection (or application start) events (Connect$\rightarrow$ Connect) in the system. This event was not an interaction within the visualization but it signaled the start of a session. What was out of place to the participant was that about 40% of sessions (137/338) had more than one connection event. This appearance did not make sense so the participant specifically explored its prevalence. It might have been caused by a logging error, which was useful information to an interaction log analyst.

  The capability to allow users to upload their own datasets was clearly an important feature as evidenced by its amount of usage in the chart. However, selecting one of the sample datasets, CS Faculty (Sample dataset 2), that showed background information of the CS faculty surprisingly garnered even more interest from the users. This behavior may be because List View was designed in an academic research environment so the visitors of the application may have had some more interest in this dataset.

- CiteVis

  The patterns explored in List View were charted over sessions and lengths in Figure 61. One key observation is that the obvious outliers in this chart, the labeled ones, are mostly related to authors, even though the participant also explored the use of other properties of publications. This observation showed the importance of authors to the users of CiteVis.

- Office Visualizations

  The patterns explored in Office Visualizations were charted over sessions and charts in Figure 62b. Similar to the CfD Study charts (Figure 51), I expected that the number of sessions and charts were strongly positively correlated. Because each chart could be used in multiple sessions, if a pattern was only used extensively in a small set of charts, the pattern would be located below the diagonal trend line. But as expected, all patterns seemed to fall neatly around the diagonal trend line, indicating that they were about evenly used in different charts. "Generate failed tree" was the most used pattern. A failed tree was a tree without a root node in the log data. This observation is disappointing because that means many users seemed to be having trouble in successfully generating tress. From the bar view in the Explore view, the participant observed that many of the occurrences of these failed trees seemed to happen early in a session, indicating that many users might have been initially inputing data incorrectly. This observation could mean the UI was not intuitive enough to users to successfully generate a tree on the first try. Furthermore, a significant amount of short sessions seemed to had ended upon a failure to generate a tree, indicating the obstacle it posed. But on the bright side, many users tried again to create a new tree as indicated by the frequent "Generate failed tree→bindFromPrompt" sequence in Figure 62a.

Another observation is that the amount of interesting trees, which were successfully generated trees with more than a depth of 1 and not a sample tree (max depth: 7, leaf count: 31), was about the same as the amount of boring trees, which were trees with at most a depth of 1. From the Explore view, it seemed that many interesting trees were generated from a sample dataset. Therefore, perhaps users were only comfortable creating simple treemaps with their own data. This type of simple treemap is now supported in Office 2016.

Pattern Assessments

One participant for each visualization application saved a number of patterns and answered a set of questions for each pattern (Table 14b). Some patterns were considered particularly unexpected and/or insightful. In this section, I present findings of a subset of these patterns that had these properties.

- VISLists

  Overall, the participants did not discover any significant insights when analyzing VISList's interaction data (Table 14b). However, one participant did notice a few patterns having unexpected amounts of usages. For example, the "Clear selection" event was used more than expected (Figure 59, Figure 68). A user can click a "Clear" button above each list to clear the selections in the list. The participant was surprised to find this pattern as being one of the most used features because a user could easily clear the selection by simply selecting another list item. The selection would achieve a main goal for clearing a list—to make a new selection. One reason for this finding was that the label "Clear" might have been misleading. VISLists was a modified web version of Jigsaw's

List view [44]. But in Jigsaw's List view, the same button has a different function—removing all items displayed in a list instead of only removing the selections. Jigsaw allows a user to selectively display a subset of list items using search or co-occurrence with items in a neighboring list. To select a subset, the first step is to clear the list if it is not empty. Therefore, a user may repeatedly use the Clear button to find different subsets of items. However, in VISLists, the feature for selecting a subset of items did not exist anymore. As a result, the "Clear" button became only for clearing selections. Because VISLists was developed by the same research lab, I would not be surprised if many users of VISLists were users of Jigsaw. As a result, these users might have tried to use the Clear button for a Jigsaw feature that did not exist in VISLists.

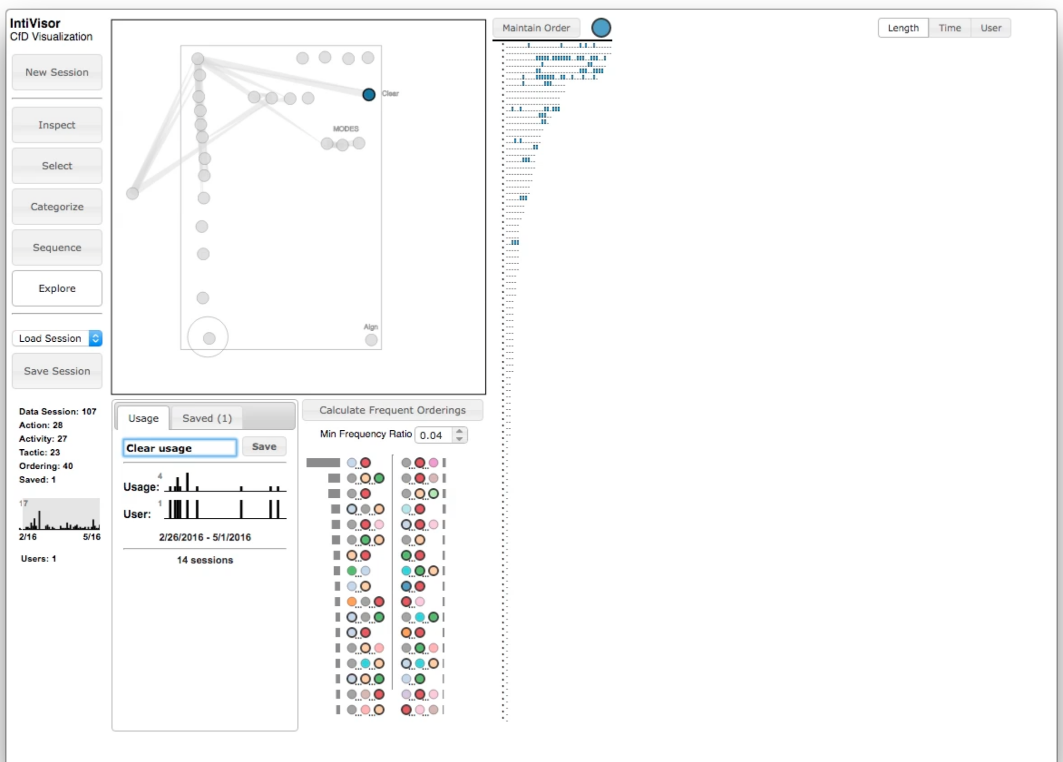


Figure 68: "Clear selection" of VISLists selected (labeled "Clear") in the Explore view. Note that the visualization label on the upper-right corner was incorrectly labeled "CfD Visualization."

- List View

  Three highly insightful patterns were saved from the analysis of List View (Table 14b). The first pattern was the change of font size (Figure 69). A significant amount of font size adjustments were discovered in the Select view of IntiVisor. One participant was surprised by the pattern because changing font sizes should not be a highly used feature. Was the default font size inappropriate? Upon further analysis in the Explore view, it turned out that these events mostly occurred in two sessions. In these sessions, perhaps one or two users were trying out the feature or figuring out the optimal font size for an application. The combinatory use of visualization views in IntiVisor helped uncover a more complete picture on how the feature was used. If these visualizations were not available, one might conclude that something was wrong with the default font size.

  The remaining two patterns were both related to an important feature—uploading data. The key difference between List View and VISLists was the capability for users to upload their own data. As a result, one participant who developed this feature was particularly interested in observing how people were using it. The first observation was that a large number of sessions seemed to end with an upload. This behavior indicated that perhaps the users were not able to successfully upload their own data. The uploads were known to fail due to a memory limitation when the file a user attempted to upload was over a certain size. Unfortunately, this error was not logged by the participant so it could only be speculated at the time of the study. Similarly, the second pattern includes consecutive upload events that indicated a user might have retried an upload due to an error. Therefore, the participant indicated in the questionnaire that these patterns could indicate a design issue. The discovery of these patterns were considered particularly insightful because the participant was very concerned about these issues.

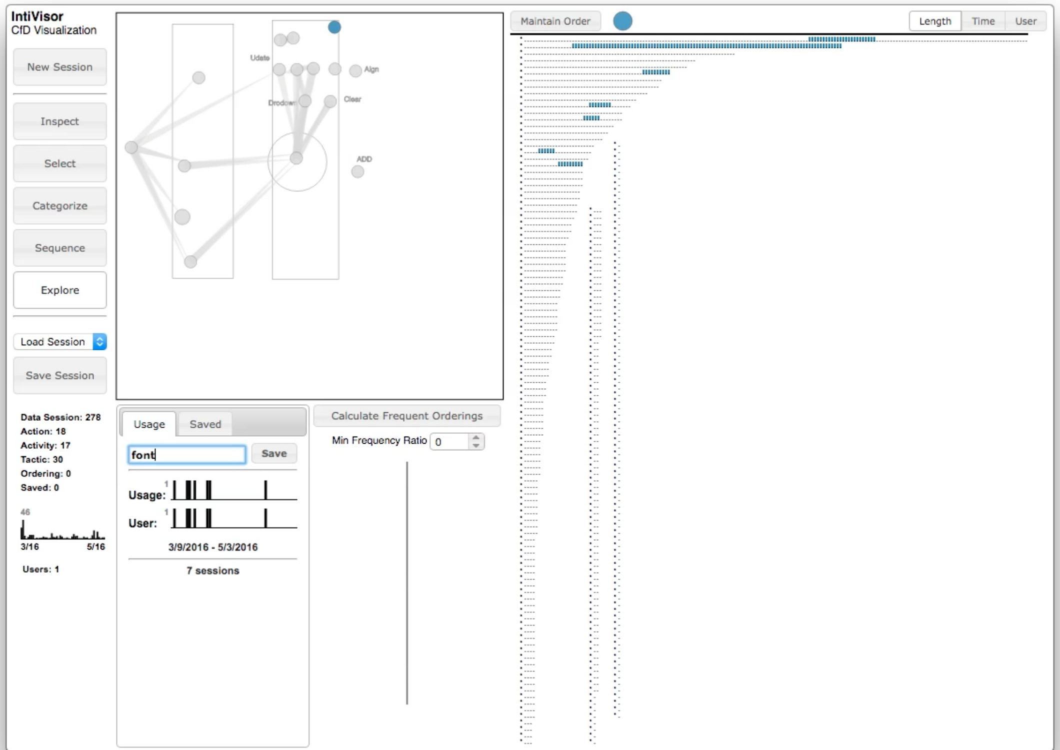


Figure 69: Font size adjustment of List View selected (unlabeled) in the Explore view. Note that the visualization label on the upper-right corner was incorrectly labeled "CfD Visualization."

- CiteVis

  Two patterns discovered in CiteVis were considered very insightful (Table 14b). One is the identification of when a new property of papers—affiliation—was added to CiteVis. The affiliation data were not originally available in CiteVis. They were added after interactions began to be logged. But at the time of the study, the participant could not recall the time when they were included. Using IntiVisor, the participant was able to identify when the affiliation data were added to properly interpret their relevant features' usage.

  The other pattern that was rated highly insightful was the ordering "Dropdown

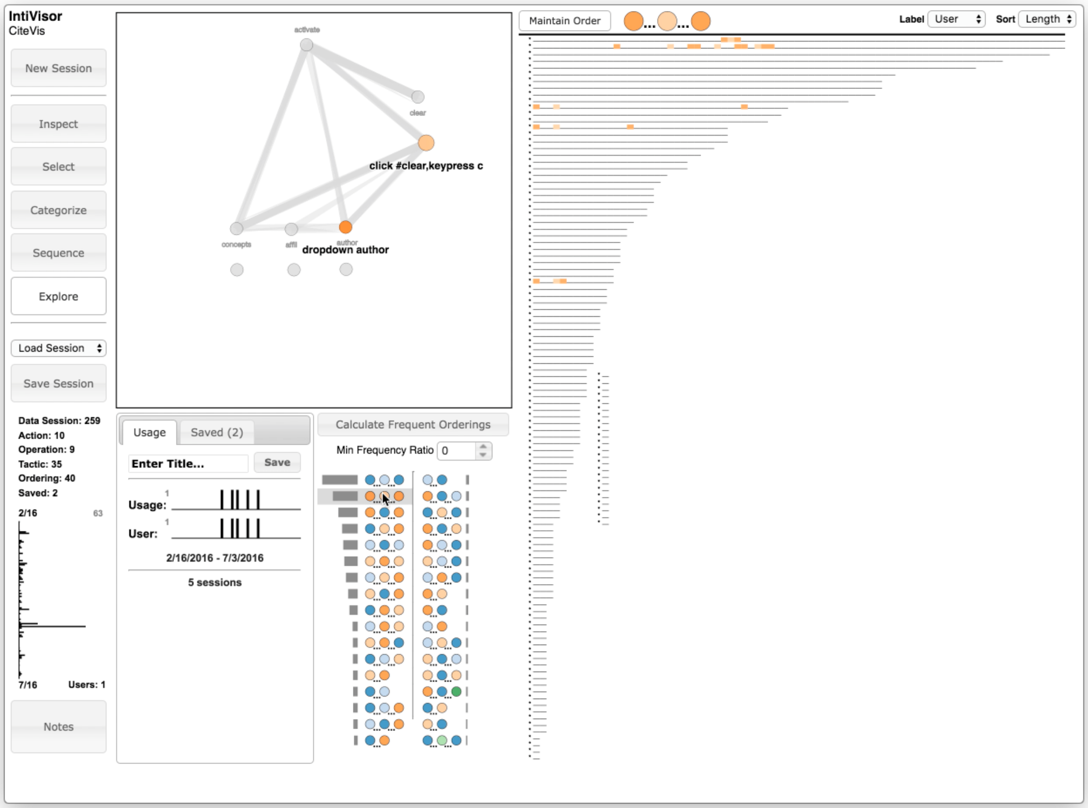


Figure 70: "Dropdown author...Clear (highlighted papers)...Dropdown author." selected from Frequent Orderings view in the Explore view. It was the second most frequent ordering.

author...Clear (highlighted papers)...Dropdown author" (Figure 70). This ordering indicated a user might have been interested in exploring papers from different authors. The participant found this pattern to be particularly insightful because it was automatically identified by IntiVisor in the Frequent Orderings view as the second-most occurring ordering in the interaction data. Based on the pattern, the analyst further examined how frequent similar patterns were with other properties (affiliation and concept) of papers. The discovery of this pattern exemplified how the automated pattern discovery algorithms could help extract insightful and meaningful patterns from the dataset.

- Office Visualizations

Several patterns discovered in Office Visualizations were considered valuable (Table 14b). For example, the participant inspected the generation of a variety of trees together as a pattern: failed trees (root count: 0), boring trees (max depth: 1), sample tree (root count: 7, leaf count: 31), and interesting trees (all other trees) (Figure 71). This pattern, titled "Learning," had a high level of unexpectedness and insight. Several observations from these trees emerged. For example, many sessions have boring trees, indicating that perhaps a simple tree was sufficient for many usage scenarios. In addition, the first tree generated in many sessions were failed trees and a large number of these sessions ended with the failed tree. These observations indicated that the UI might need to be improved because users were failing to successfully generate a tree on their early attempts.

Another pattern ("playwithdata") that was of interest to the participant was observing when users were playing with data. The participant had little knowledge about the pattern and acquired a high level of insight. The pattern showed that a user loaded data (load) and then repeatedly edited the data table (readBoundData). The participant noted that these users chose to update the visualization by changing the data in a bounded data table instead of changing the data table bound. This behavior was unknown to the participant in the first place so it was of particular interest.

C. How do we provide analysts the capability to practically identify and analyze patterns?

The average effort rating for identifying the saved patterns was 1.15 on a 3-point (1-3) scale with 1 being of the lowest effort (Table 14b). This rating was much lower than that in the CfD Study where the average was 1.83 (Table 8b). These ratings indicated that the system did seem to be easier to use for analyzing simpler visualization applications and to these more technical participants.

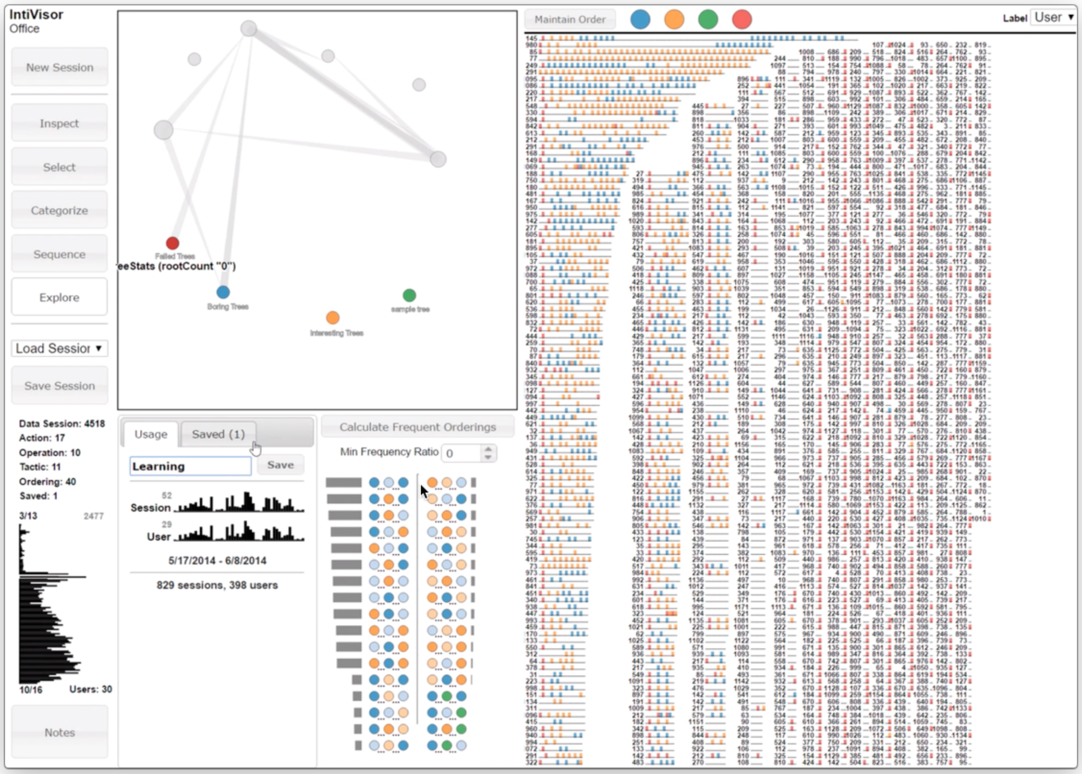


Figure 71: Failed trees (root count: 0), boring trees (max depth: 1), sample tree (root count: 7, leaf count: 31), and interesting trees (all other trees) of the Office Visualizations selected in the Explore view.

One participant of each visualization application instrumented the applications to log interaction events. The participant who logged interactions in both VISLists and List View considered the logging process to be very simple because it was only about adding a few lines of code in the application. But the participant who logged interactions from CiteVis had a different opinion, indicating the logging effort to be the most challenging obstacle to adoption. This participant considered the difficulty in determining what to log and the amount of details to log. The challenge is that typically the realization of insufficient logging occurs during analysis, which is after the logs have already been collected. The solution to the problem is to frequently analyze and refine the interaction logs to ensure that they are collecting enough useful information. However, this solution introduces an even higher overhead that typical designers may not wish to bear.

The participants also indicated the most challenging aspects in using IntiVisor. For example, two participants indicated that the selection of patterns in the Explore view to be challenging. From the observations, it was clear that the UI for selection was not very responsive so that sometimes when the amount of data was larger, it could take a while for the selection to complete. Because of such performance issues, the system included features such as sampling to limit the amount of data being analyzed at a time. One other participant, who always preferred to see more data, did not encounter such performance issue and commented that the system seemed to be limiting the amount of data being visualized too much for performance. Therefore, these feedbacks indicated a performance improvement of the system is needed to help streamline the analysis process. Another challenge was in the categorization process. One participant in an early study indicated that the dragging and dropping of actions to categorize them into operations as one of the least practical aspect in using the system. The participant suggested an automated operation assignment and layout, perhaps based on the most likely occurrence sequences of those operations. The feedback was addressed using a simpler automated assignment that laid out operations in a grid. However, in later studies after the feature was implemented, one participant chose not to use the new automated assignment feature because it might assign actions into "weird places" on the canvas. This response indicated that some analysts prefer to take more control over the perspective generation and action assignment process, even though it was more labor-intensive. Another participant thought a circular layout could work well. The participant created a perspective using a mostly circular layout during the study (Figure 67).

5.4.7 Conclusion

In this study, I applied the visual interaction analysis framework to three other visualization applications. These applications were simpler than the CfD visualization and the participants all had more technical backgrounds having built their visualization applications. As a result, I found that not only all the participants were able to operate IntiVisor by themselves but also perceived the process to be less labor-intensive when compared to the CfD participants. Overall, the participants were able to utilize the flexibility in the system to create their own perspectives and to identify a set of patterns and insights for each visualization application. This study preliminarily demonstrated that the visual interaction analysis framework could be successfully applied to other visualization applications.

Table 14: Saved patterns with questionnaire responses of the Generalization Study. (a) Numerical and color-coding in the questionnaire table. The questions were presented to users in this order in a UI dialog. The colors are chosen semantically to map to more interesting responses. (b) Questionnaire responses are organized in separate columns. Notice the Indication of Design Issue question was moved to the far right instead of in its original dialog position because it uses a different scale (0-2 instead of 1-3). Because the colors of responses are mapped semantically, an analyst can examine the spread of green to determine which patterns are more interesting to a participant. For example, the "Upload problem" pattern in List View was considered valuable because of the spread of green in many aspects based on the spread of green in the responses. On the other hand, the mostly green responses in the Level of Effort column shows that participants in this study mostly considered the pattern discoveries to be relatively effortless.

1	Knowledge of existence	1	Low	2	Medium	3	High
2	Unexpectedness of usage amount	1	Low	2	Medium	3	High
3	Indication of design issue	2	Yes	1	Maybe	0	No
4	Insight significance	1	Low	2	Medium	3	High
5	Amount of effort	1	Low	2	Medium	3	High

(a)

Application	Title	Level of Knowledge for Existence (1-3)		Level of Unexpectedness in Usage Amount (1-3)		Level of Insight (1-3)		Level of Effort (1-3)		Indication of Design Issue (0-2)	
VISLists	Multi confusing	Medium	2	Medium	2	Medium	2	Low	1	None	0
	Clear usage	Medium	2	Medium	2	Medium	2	Low	1	Maybe	1
	Dropdown change	Medium	2	Medium	2	Medium	2	Low	1	Maybe	1
	Frequency usage	High	3	Low	1	Medium	2	Low	1	None	0
	By connection	Low	1	Medium	2	Medium	2	Low	1	None	0
List View	font	Medium	2	Medium	2	High	3	Low	1	None	0
	Upload problem	Low	1	High	3	High	3	Low	1	Maybe	1
	Multiple uploads	Medium	2	Medium	2	High	3	Low	1	Maybe	1
	Order by frq	Medium	2	Medium	2	Medium	2	Low	1	None	0
	Connect to tool	Medium	2	High	3	Medium	2	Low	1	Maybe	1
CiteVis	affil appears	Medium	2	Medium	2	High	3	Medium	2	None	0
	care about auths	Medium	2	Medium	2	High	3	Medium	2	None	0
	care about affil	Medium	2	Medium	2	Low	1	Low	1	Maybe	1
	care about concp	Medium	2	High	3	Medium	2	Low	1	None	0
	auth-paper	High	3	Low	1	Low	1	Low	1	Maybe	1
Office Visualizations	Education test	High	3	High	3	Medium	2	Low	1	Yes	2
	Learning	High	3	High	3	High	3	Low	1	None	0
	loadsave	Medium	2	Medium	2	Low	1	Low	1	Maybe	1
	loadchange	Medium	2	High	3	Low	1	Low	1	None	0
	playwithdata	Low	1	Medium	2	High	3	Medium	2	Maybe	1
Average	-	2.05		2.20		2.15		1.15		0.55	

(b)

CHAPTER VI

CONCLUSION

In this thesis, my contribution is in designing an visual interaction analysis framework that provides flexibility and practicality in the analysis process for extracting patterns from user interactions. The framework includes a visual interaction analysis system, IntiVisor, that supports the analysis process with visual analytics. I evaluated the framework via the analysis of interactions logged from five visualization applications: CfD Visualization, VISLists, List View, CiteVis, and the Office Visualizations. The evaluation showed that the framework was able to provide flexibility in the analysis process for identifying useful and meaningful patterns when used by analysts with technical backgrounds to analyze interactions from visualization applications of low to medium complexity. The success of applying the framework to multiple visualization applications demonstrated the generalizability of the framework. In this chapter, I further discuss the benefits of the framework as well as its limitations. I also outline opportunities for future work.

6.1 Discussion

In this section, I discuss my observations of applying the framework, including general limitations and challenges that arose in this work.

6.1.1 Application to Interactions of Complex Visualization Applications

Although the framework was applicable to multiple visualization applications, these applications were generally of low to moderate complexity. The CfD Visualization was the most complex, with three visualization views but only one view—the Timeline view that supports basic chart types (line chart/bar chart/scatterplot)— was extensively used (Appendix A). VISLists and List View were a simpler, web-based version of the List view in Jigsaw [44]. That view is only one of the eight visualizations available in Jigsaw. As a result, it was unclear how the framework would apply to a complex visualization application that has significantly more views and interactions like Jigsaw itself.

In general, I believe that IntiVisor could be useful in analyzing more complex applications, but clearly some challenges may arise. For example, complex visualizations could have multiple views that may not appear at the same time. In Figure 28, I showed how perspectives could be generated for such complex visualization applications by mapping all relevant views onto a 2-D canvas. Complex visualizations could have a large number and variety of interaction events. The amount of these events should not be a problem for the Select and Sequence views of IntiVisor because they are highly scalable as data in these views are sampled. But a limitation can be reached in the bar view of the Explore view where more complex visualizations may have interaction sessions that are longer in duration and have more interaction events. In these cases, analysts may need to horizontally scroll the view to examine the entirety of long sessions. To minimize the need to scroll, the width of each colored block could be reduced using a resizing feature. Nevertheless, at some point, it will still not be enough. In this case, a larger widescreen monitor is recommended to be able to leverage the additional horizontal pixels to show these long sessions. Furthermore, complex visualization could have more interval-based interactions that may need to be logged and visualized with its duration. IntiVisor was designed for point-based interactions so to support interval-based interactions, such as dragging items or using gestures, it might need new visual representations.

Up to this point I have been discussing visualization applications used by one person at a time. However, another type of complexity in a visualization application is when multiple users can collaboratively interact with it. In these cases, a single session may have more than one user and the visualization can be operated by more than one user at the same time. Because of such multi-user usages, the timestamps of the events will need to be very accurate and synchronized between user machines to determine the turn-taking patterns and what each user is able to see at the time of their interactions. IntiVisor was designed on the assumption that each session would only have one user. That is why each horizontal bar in the bar view can only map to one user (Figure 38a). The system would need to be modified to accommodate this type of multi-user collaborative applications.

6.1.2 Regular Expressions

As discussed in the Field Study (Section 5.3), IntiVisor did not support specifying patterns that did "not" include a specific interaction or interaction sequence. For example, an analyst could not specify a pattern such as "Zoom In"... ~~"Zoom Out"~~ ..."Pan Left" that indicates a user panned left some time after zooming in but did not zoom out at any point between these interactions. Moreover, IntiVisor does not explicitly support selecting alternatives such as finding Zoom "or" Filter as part of a pattern. But there is a workaround. An analyst could use the Categorize view to categorize Zoom and Filter as a new category first and then select this new category in the Explore view. The problem is that Zoom and Filter by then will not be differentiable anymore. The ability to directly select such patterns is currently lacking in the system.

These patterns can be very important. A generalized need of identifying patterns like these is to be able to search for any type of pattern with regular expressions. The challenge is that the UI for compiling these complex patterns could also become complex. Seeing that IntiVisor was already difficult to use for some analysts, I would recommend avoiding the support of rarely investigated patterns in the main visualization UI and suggest providing a dialog somewhere to support advanced analysts in specifying complex and rarely explored patterns with advanced regular expressions.

6.1.3 Overhead in Logging Interactions

From the study, I received conflicting opinions about how laborious and challenging it was to instrument a visualization application to log its users' interactions. In Section 4.2, I outlined a couple of methods for instrumenting applications and recommended the participants to use the methods. However, the methods did not seem sufficient. The primary challenges were in deciding which interaction events should be logged and how much detail should be logged. These challenges could not be easily reduced with the recommended methods and are concerning because the participants were instrumenting visualization applications that were not considered very complex.

One idea of reducing the instrumentation labor is to use a catch-all event handler that automatically logs all the application-wide interaction events automatically with their default parameters. The problem with this logging mechanism is that the logged events may be even less interpretable because an analyst did not make the decision on their individual formats and contents. This interpretation effort is then transferred to the analysis process. Furthermore, a catch-all event handler could log significantly more data about an event that are not all useful to an analyst, such as the coordinates of a button click, or worse, not log enough information that is crucial, such as the zoom level of a "Zoom In" event. As a result, this type of logging could save time during the instrumentation process, but may easily lead to more effort and fewer insights in the analysis process. As a result, I still recommend analysts to log interaction events manually. The process could also help an analyst reflect on the features of an application and begin to form hypotheses on the usage patterns. The ultimate motivator for the logging instrumentation process might just be to show

how the logged interactions could be analyzed to reveal insights. When the benefit of subsequent analysis could significantly outweigh the cost of the instrumentation effort, an analyst would be more motivated to provide this effort.

6.1.4 Logging Recommendations

From the studies, I learned a few additional things about how logging could have been conducted to better support the analysis process. Typically, problems in logging only surface during the analysis process when the information in the logs is examined. Therefore, based on the studies and my analysis experiences, I provide the following recommendations that could be helpful to future instrumentation efforts.

- Log enough (manipulation, target, parameter) information to uniquely identify an interaction event

  The first recommendation was mentioned earlier in Section 4.2. Each event needs to have enough information for it to be uniquely and independently identifiable. At the very least, the manipulation, target, and parameter of an event need to be sufficiently logged. When they are not, the logged events will carry less information and become less flexible to analyze. For example, sometimes an analyst may choose to only log certain categories of events, such as zoom and filter, that seem enough to address the original analysis goals during the logging instrumentation process to save effort. When categories are used to log events, interaction events would not be uniquely identifiable when each category is mapped to more than one UI interaction. For example, if two UI components for zooming exist, such as clicking a Zoom in button and selecting a Zoom In menu item, then only logging Zoom for both of them makes it impossible to differentiate the two interaction events during the analysis process. The problem is that if new analysis goals emerge during the process that require the differentiation of these events, the goals would not be achievable. Therefore, my recommendation is to avoid only logging events that are not uniquely identifiable, such as abstracted categories. Categories should be generated during the analysis process as presented in the visual interaction analysis framework.

- Log contextual interaction events

  An analyst's first attempt at logging interaction events may be only logging interactions that seem "interesting." However, this approach is problematic because of two reasons. First, initially uninteresting interactions can become interesting during the analysis. Second, seemingly uninteresting interactions can provide valuable context to those interesting interactions. For example, if an analyst was only interested in color encoding events and only logged them, it would be difficult to know if two back to back color encoding events actually occurred consecutively. Therefore, oftentimes uninteresting events, such as a font size adjustment event, could provide a context to interesting events by revealing a fuller picture of the entire sequence of interactions. The fuller picture can lead to a more confident interpretation of usage patterns. As a result, log not only interesting events, but also seemingly uninteresting events. The goal is to log enough events to be able to recreate as much as possible the original interaction sequence.

- Log system actions when necessary

  In this work, I did not separate system (application) actions from user actions. But these system actions at times need to be separately logged to provide a clear picture of a user's experience for the best interpretation of the corresponding actions. Figure 72 illustrates three types of user interactions with a system that sometimes require the separate logging of system actions. Figure 72a shows the typical user-initiated interaction (Click Zoom In button) with an expected, corresponding system response (system action: Show zoomed in view). In this case, simply logging the user action, Click Zoom In button, is sufficient to convey both the user's and the system's actions. However, in Figure 72b, a system error, such as the view cannot be zoomed in anymore, resulted from the user action. Therefore, instead of showing a zoomed in view as expected, the system returns an error message. In this case, only logging the user action is not going to be sufficient to document the entire back and forth interaction because the system is not responding according to the user action.

One way to log this error message is as a parameter of the user action event, for example, Click Zoom In button (error: reached max zoom level). The benefit of this logging approach is that it is clear which user interaction caused the error message. But the problem with this logging approach is that only one timestamp would be recorded for both the user action and the system response. Therefore, during the analysis, there will be no way for an analyst to measure the time it took between the user action and the system response, which is valuable information for diagnosing the severity and cause of the error when this delay matters (e.g, data load time). As a result, my recommendation is to log this error message as a separate, system action event. The connection with the user action that resulted in the error could be paired during the analysis phase.

Yet another type of interaction is shown in Figure 72c—when a system initiates an action that is not connected to any specific user action. This type of initiation could be caused by time (e.g., timeout) or something else going on internally within the system (e.g., out of memory error, data change). In summary, for the second and third types of interactions, system action events are crucial to separately log because they impact the users' experience and their proceeding actions.

To implement such system action events, a new mapping is needed for the

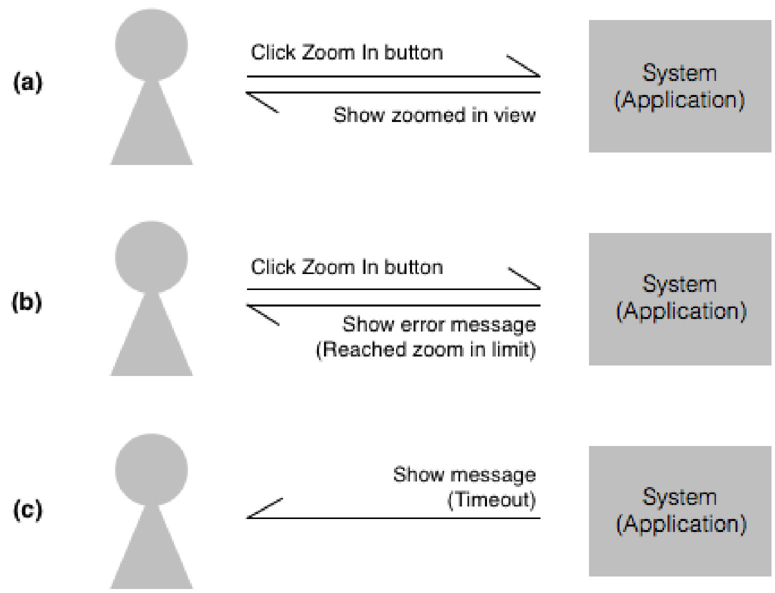


Figure 72: Three types of user interactions with a system that sometimes require the separate logging of system-initiated interactions. (a) Typical system response that corresponds to the user action (Click Zoom In button). (b) Unusual system response of an error that should be logged separately. (c) System-initiated interaction that does not map to any specific user action.

logging format as shown in Table 15. Compared to the user-initiated event logging format in Table 2, the manipulation and target fields do not apply anymore because the event is not generated by a user. Therefore, for system action events, the "user" of the interaction needs to become the "system." For example, the error message in Figure 72b would be constructed in the way shown in Table 15. The manipulation is "show" by the system, the target is "error message," and the parameter is "reached zoom in limit," the content of the error message. Having this type of system action events in the log data will immensely help the analysis process.

- Log output state as parameters

  Typically, the state of an application after an interaction is logged as a parameter in the interaction event. For example, if a user clicked a toggle button that shows or hides a trend line, the outcome of the interaction (whether the

Table 15: System action events can be logged in this format.

Manipulation	Parameter	Target	Time	User	Other
show	reached zoom in limit	error message	01-01-2015 00:00:00	-	session

trend line is shown or hidden) should be logged as a parameter in the toggle event in order to differentiate the two toggle modes. For certain visualization interactions, the state change can be more elaborate after an interaction. For example, if a user generated a treemap in the Office Visualizations, the parameters included an entire set of statistics of the tree (e.g., root count) generated from the interaction event. These statistics are very useful during the analysis as shown in the Generalization study to learn about the variety of trees (e.g., boring trees) generated from users.

Depending on the analysis goal, an analyst may choose to log the updated state of the visualization upon every interaction. For example, if the analysis goal is to determine how the size and shape of a graph changes over time in a graph visualization, the size and shape of the graph should be saved as parameters upon every interaction that changes them. This additional level of logging can significantly increase the amount of log data so an analyst should carefully decide how often the state of the visualization is logged. Alternatively, an analyst could choose to only log the state "change" from every new interaction event. For example, when a new node is added to a graph, only this addition (node count + 1) is saved in the parameter. Only logging the state change could significantly reduce the overhead in the log but the problem is that each state at any given time point will need to be reconstructed during the analysis from the event sequences. This could be a difficult task depending on how complex the visualization is and how many state changes were there.

- Log within system capability of both client and server

  Logging data remotely can cause significant overheads to both the client and the server. If the log data are collected remotely on a client machine and are sent to a server over the Internet, then the logging mechanism needs to be efficient enough so as not to cause a performance issue or a significant network overhead to the client's machine. For example, if the mouse position and the entire visualization state are logged every millisecond, it could cause a significant burden on the resources of a client machine. On the server side, the amount of log data could be large so a data storage solution that can properly handle the speed of incoming log data and the efficient retrieval of them for analysis is essential.

- Revise logging iteratively

  Similar to a visualization application, interaction logs need to be updated when errors are found or more information is needed to answer certain analysis questions. Oftentimes analysts wonder what they should log. It is considered one of the most challenging aspects in the logging process simply because it is almost impossible for an analyst to know all the information that is important or useful until the analysis phase. As a result, a more practical way of logging is to periodically and iteratively analyze log data and revise information being logged when necessary. This periodic analysis is only feasible when an effective interaction analysis system is available such as IntiVisor. When logs are updated, they need to be properly documented so that later analyses can consider this change.

6.1.5 Challenges in Scalability

Interaction logs can accumulate quickly so scalability in analyzing them is a crucial challenge. IntiVisor addresses the scalability issue by sampling (e.g., Figure 22) and selecting (e.g., Figure 17d) data. These mechanisms work well in reducing the amount of data that needs to be processed at a time. The benefit of using sampling and selection is that it does not matter how much data there are—the amount of data that were being processed and displayed can always be limited to a certain size. The problem is that if an analyst does not wish to use a sampled dataset, then a more effective data processing mechanism and visualization platform will need to be supplied. The algorithms to extract frequent sequences can also be slow to run locally in a browser. One potential improvement is to move the heavier sequence extraction tasks to a server that can compute faster. Furthermore, IntiVisor visualizes data in a browser using SVGs so when a large amount of SVG tags are used, the drawing and interaction performance can significantly decrease. Therefore, a different visualization platform could be developed to overcome this limitation.

6.1.6 Identifying VAMs

From my studies, I learned that it may be difficult to identify VAMs or strategies solely based on interaction logs. A VAM is a high-level abstract representation of an interaction sequence. One key property of a VAM is whether a user "planned" to use a specific method. As a result, a frequent sequence of "Zoom In"…"Adjust font size"→ "Zoom In" may show a popular usage pattern, but it is still a leap to call it a VAM without knowing whether the users planned to analyze the system in this specific way. Furthermore, are the operations in the sequence, such as "Zoom In," abstract enough to be part of a VAM? Or should a more abstracted representation such as "Zoom" or "View port change" in Figure 52 be a better unit for a VAM? A VAM should be highly composed and abstracted (upper-right corner of Figure 52) with information about a user's plan or high-level intent. Unfortunately, interaction data typically only provide enough information to determine low-level user intents (e.g., Yi et al.'s interaction categories [50]). That is why Kang et al. identified VAMs (strategies) in Jigsaw with more than just the interaction data [25].

6.1.7 Types of Insights from Analysis Tasks

From the studies, many insights addressing the analysis tasks listed in Section 3.1 were discovered. In this section, I will discuss the variety of insights found using the framework based on the interaction analysis tasks.

- Assess Usability

  Analysts identified several issues when assessing the usability of their visualization applications. For example, in the Office Visualizations, many users initially failed to generate a treemap, as shown in Figure 71. A large portion of users ended their sessions when this failure occurred. This observation indicated a usability issue where the application UI might not have been informative enough for users to generate treemaps on their first tries. In List View, the upload mechanism seemed to have some problems where many users simply left or needed to repeat their upload actions to successfully upload their datasets into the application. These usability issues caused "problematic patterns" to emerge so upon visual inspection, these patterns stood out for analysts to discover and potentially to address in the future.

- Assess Utility

  Analysts can glean the utility of features from the amount of usages. For example, some features that were almost never used, such as the Circular view in the CfD Visualization, could mean that they were not useful. On the other hand, in VISLists, certain widely features such as "Sort by frequency" combined with extensive activities on finding "mosts" (Figure 59b), strongly indicated that the features revealing the most frequently occurring items were very useful.

- Learn About Users

Analysts learned about different types of users from usage patterns. For example, in the CfD Visualization, some users always started their sessions by removing behaviors and/or medications. Some users always toggled on the night shift data. Some users had very short interaction sessions that configured the visualization into a standard format and then printed it. These different user-specific usage patterns may not directly link to usability issues or reflect the utility of the CfD Visualization, but they are valuable to analysts to learn about the variety of their users.

- Understand Usage Patterns/Analysis Methods

  Analysts explored a variety of user-independent usage patterns, from low to high levels, to understand their occurrences and prevalence. For example, for low-level patterns, an analyst of VISLists examined the use of font size adjustments. The pattern was extensively used in two sessions and typically occurred in chunks when users fine-tuned the font size (Figure 69). For higher-level patterns, the analyst of CiteVis found that multiple users likely highlighted papers from one author, at some point afterwards cleared the highlight, and later highlighted papers from likely another author (Figure 70). The discovery of this unexpected usage pattern prompted the analyst to further explore similar patterns with highlighting and clearing of concepts and affiliations in CiteVis. Unfortunately, for analysis methods, as discussed in the "Identifying VAMs?" section, it is not easy to determine VAMs from only user interaction events. To find VAMs, the users' intents will need to be considered.

6.1.8 Implications for Design

Many insights obtained from the studies led to implications for design (Indication of Design Issue: Table 8b and Table 14b). In the following, I present a list of implication categories that have been considered within the studies that show how the framework could lead to potentially actionable improvements of the user experience.

- Change Default Configurations

  A visual configuration that is frequently switched to after starting the visualization application is a candidate for becoming a part of the default configuration. For example, in VISLists, many users chose to order the list items by frequency (Figure 59b). Because of the extensive use of this view configuration, one of the participants considered using it as the default sorting mechanism. But the change of default view configurations should always be carefully considered because it will impact all users. Because a large portion of users switched to a view configuration does not necessarily mean that they did not use or consider the current default configuration useful. A simple experiment is to change the default view and then examine how often users switch back to the original default view. If more users are switching back, then perhaps the original default view was more useful.

- Create Shortcuts

  Interactions that frequently occur together could be candidates for merging into a shortcut to reduce the amount of interactions required from users. For example, users of the CfD Visualization toggled night shift data on and off possibly to compare the visualization before to after including these data. An analyst could consider merging these two events into a shortcut that automatically toggles the data on and off with a short delay in between to save a click from a user every time. However, some interactions that frequently occur together cannot be merged. For example, mouseover and mouseout events over a visual item frequently occur together. Typically, some additional information about the visual item will be displayed when the mouse cursor is hovering over the item. These two events cannot be merged as a shortcut because the events are generated from mouse movements where the location of the mouse cursor also indicates the visual item to focus on. Therefore, whether events can be merged as shortcuts depends on an analyst's knowledge about the nature of the interactions.

- Improve Performance

  Certain interactions may indicate a performance issue within the visualization application. For example, in List View, the upload feature may have failed for many users so that they left or needed to retry. This type of performance issue is important to discover and fix to improve the usability of the application.

- Redesign UI

  Some usability issues are too extensive that may require a reassessment and redesign of the UI. For example, the Office Visualizations had this problem with generating treemaps so that many users were having trouble in successfully generating a treemap on the first try (Figure 71). This type of early failure indicates that the UI was perhaps not clear enough to users. As a result, the UI for generating treemaps needs to be reevaluated to see how to improve the UI of the application so that the amount of initial failures can be reduced.

- Re-educate Users

  Features not frequently used may need to be reintroduced to users. In the CfD, the users were staff members of the center so that features that were less used can be reintroduced to the users to re-educate them about the features. For example, the bookmarking feature and the Circular view were both reintroduced in workshops to remind users about their utility and method of use (Figure 56). This re-education process helped a few users pick up the bookmarking feature.

6.1.9 Generalization to Interaction Analysis of Non-visualization Applications

In this thesis, the visual interaction analysis framework was applied to interactions of visualization applications but it can also be applied to interactions of non-visualization applications. However, visualization interactions may have a few properties that make them more interesting to analyze in the framework. For example, visualizations support the externalization of data. So when users interact with visualizations, they change the visual representation of the externalization based on their thought process. This thought process, or reasoning process, is pronounced in visualization interactions because usages are typically exploratory in nature. Therefore, by capturing visualization interactions, an analyst may be able to uncover the internal reasoning processes of users [12]. This information is very valuable for evaluating visualization applications and understanding users.

6.2 Looking Forward

Visualization interaction analysis using visual analytics is beginning to gain traction in the visualization community. The new "Logging Interactive Visualizations & Visualizing Interaction Logs" workshop at this year's VIS conference is a forum specifically for discussing topics surrounding this line of work. The emergence of this workshop shows that more visualization researchers are becoming interested in the vast amount of information that could be extracted from analyzing interaction logs using visual analytics.

A visual interaction analysis framework provides a guideline for analysts to systematically examine interaction logs to discover patterns and insights. IntiVisor is a proof of concept on how the visual interaction analysis system in the framework could be implemented to provide flexibility and practicality in the analysis process. But the system has plenty of room for improvement to be applicable to a wide range of interaction formats and scalable to handle large amounts of data. Therefore, future researchers could use my experience to build an even more flexible and practical visual interaction analysis system. I hope this work can help advance the interest and adoption of visualization interaction analysis so that in the future, interaction log analysis could be more extensively utilized to study visualization applications.

APPENDIX A

CFD VISUALIZATION

A.1 Background

The Center for Discovery (CfD) is a school and residential program for the care and treatment of people, most of them children, with significant medical complexities and developmental issues. These people, whom I will call students, may have behavioral issues such as aggression. To eliminate or reduce these problem behaviors, the CfD staff extensively track the problem behaviors and their potential causes, such as medication, sleep, and bowel movements, of the students. However, with all these data collected, they need an effective way to analyze the data and communicate findings.

The CfD has 25-30 Behavior Specialists and Behavior Analysts (whom I will collectively refer to as BAs) whose primary responsibilities are to collect, aggregate, and analyze these data. These BAs are case managers of students. They need to organize data collected from students and present them to the care team to determine treatment plans. The care team typically consists of staff members from different departments of the CfD (e.g., psychology, residential, nursing) that are taking care of the students. In the past, BAs used Microsoft Excel to visualize some of the data for this purpose but reached several limitations of its static charting tools. For example, it was not possible to add or remove variables being shown or rescale views once the charts were generated. Therefore, if someone viewing a chart would like to load more data, remove some visual items, or zoom the view, the BAs would not be able to provide the new charts on the spot easily. This limitation poses a problem during their discussions at team meetings where they often need information from these charts to make critical treatment plan decisions.

A.2 Visualization Design

To solve the problem, a team of CfD staff and I collaboratively designed and deployed an interactive visualization application to support the analysis of these data with the BAs as the primary users. The following high-level design criteria were established to minimize the impact on the practice and care of the students.

- Leverage familiarity

  The visualization should leverage the BAs' familiarity of charts that the CfD used in the Excel spreadsheets to lower the learning curve. We started out using the same line and bar chart representations with which the BAs were familiar and later moved to more complex visualization techniques that can reveal different types of insights.

- Minimize overhead

  The visualization should minimize additional workload. Ideally, the workload of the BAs should be decreased. We designed the application to support data import from Excel workbooks with which they had been collecting data. Importing the data into the visualization application was as simple as dragging and dropping a file into the application.

The visualization application was built on a web browser and has several views: Start page (Figure 73), Timeline view (Figure 74), Parallel view (Figure 75), and Circular view (Figure 77). The data in these views were made up for illustration. I next present an overview of these views.

A.2.1 Start Page

When users access the application, they first encounter the start page. It allows the user to import data, consent to our research study, and see the latest updates and frequently asked questions. See Figure 73 for the start page.

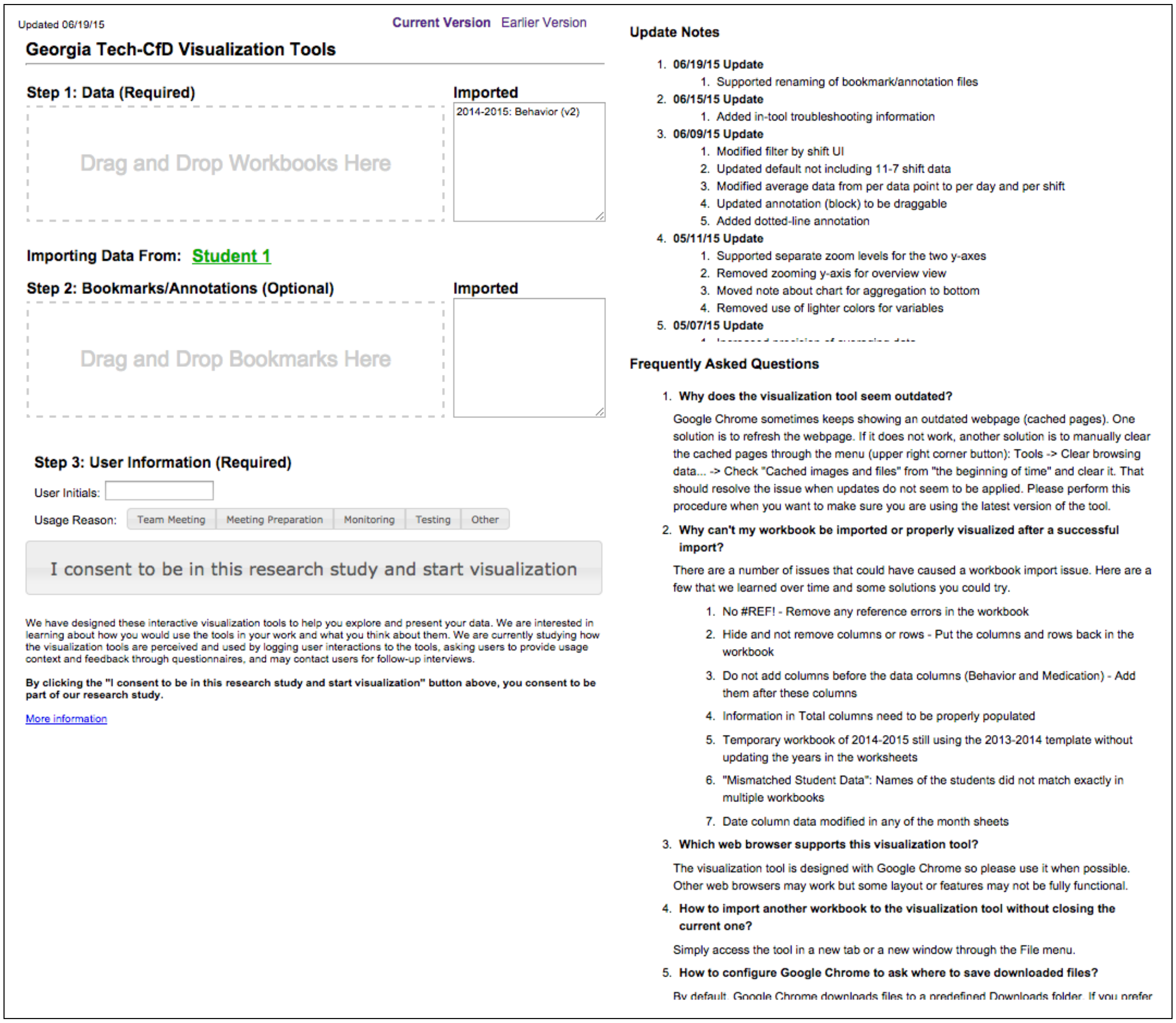


Figure 73: Start page of visualization application

A.2.2 Timeline view

The Timeline view shows data on a timeline similar to the charts the CfD staff previously used in Excel. Data can be visualized in lines, bars, and circles. Figure 74 shows the visualization of data from two variables: one visualized as lines and one as bars. The view supports the typical interactive visualization features that the CfD needs such as zooming and filtering. It also supports other features that are specifically useful to the CfD, such as multiple methods of data aggregation (e.g., average data by month, sum data by day). Additionally, it supports bookmarking and annotating views.

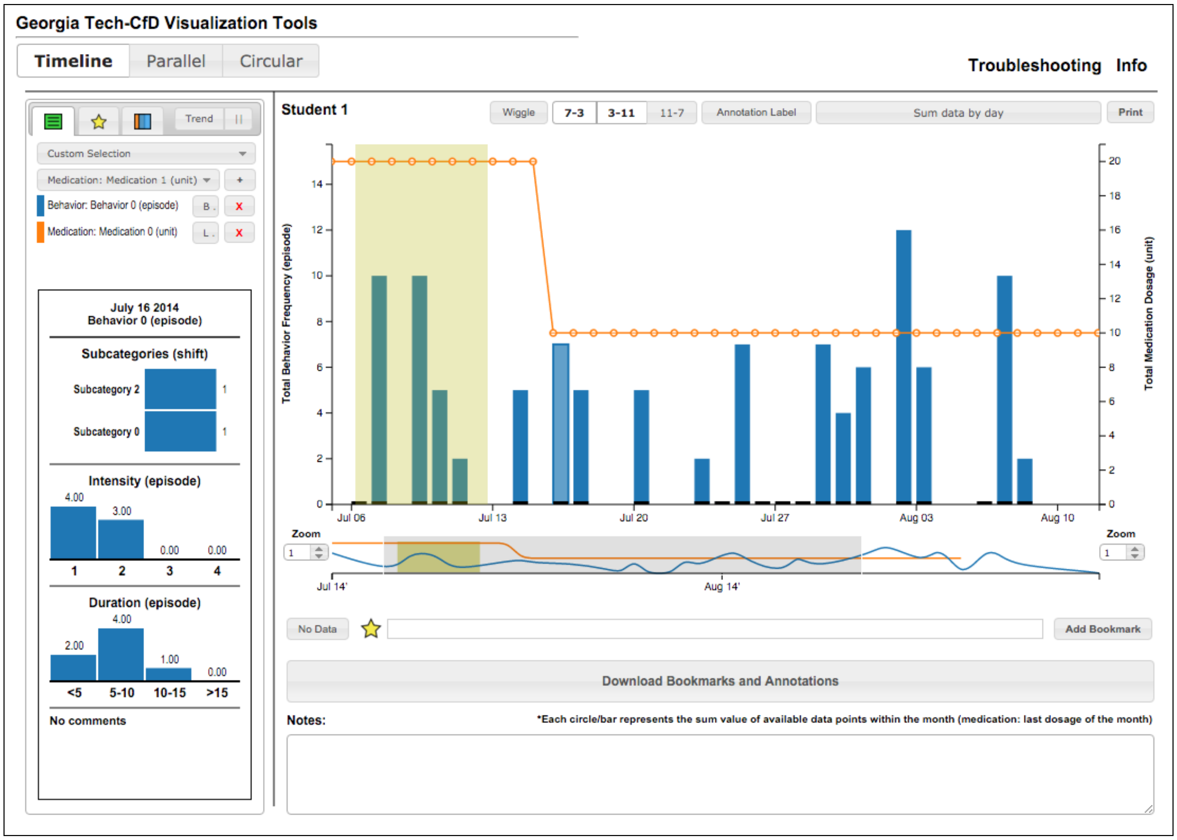


Figure 74: Screenshot of the Timeline view. A data point (bar) in the middle is selected to show additional information in a panel to the left. An annotation (yellow rectangle) is created in the left of the view.

The Timeline view was deployed in May 2014. I visited the CfD for a set of presentations and workshops. The presentations were given to the BAs and other CfD staff that may view the visualizations. The workshops, which provided tutorials on the features, were only for BAs that would be using the visualization application.

The initial deployment of the visualization application prompted a number of work practice changes at the CfD. For example, they redesigned their workbooks to collect a higher granularity of data so that more information could be visualized in the application. A new version of the application was deployed a few months later to support this new format. Moreover, the CfD decided to stop providing static chart templates in their redesigned workbook to encourage the adoption of the new visualization application. Both these changes implied that the visualization application was valuable and effective enough to support their needs.

A.2.3 Parallel view

The Parallel view uses the parallel coordinates visualization [23] to show the relations between a set of variables when their data points co-occur in a given time window. For example, this view can answer questions such as, "on the days when a student slept less, what was the frequency of the student's aggressive behaviors?"

Figure 75 shows the Parallel view. The view includes the two variables selected as the axes to the right and one additional temporal axis to the left. Each axis includes a range of values that map to the minimal and maximal values of the variable in the loaded dataset. For example, Behavior 0 has a range of 0-8 episodes. Each line connecting the axes represents a shift, day, or month, depending on the configuration above the view. If a line is a shift, as shown in Figure 75, its connecting points on the axis indicates the variable value at that given shift. For example, in the shift highlighted (red), 5 episodes of Behavior 0 occurred. The numeric value at the bottom of each axis by default shows the number of data points loaded for that variable.

Each axis can be interactively brushed to filter the connecting lines. When an axis is brushed, all lines that connect to the axis outside of the brushed value range are filtered out. For example, in Figure 75, a brush is applied to Behavior 0 that indicates only shifts (lines) that have over 5 episodes of Behavior 0 are to be kept in the visualization and others are temporarily filtered out. This interaction can let the CfD staff interactively find the answer to the question "when a higher amount of Behavior 0 occurred, when did they occur (map to time axis) and which Medication 0 dosages were administered (map to Medication 0 axis)?" When an axis is brushed, the numeric value at the bottom of the axis is updated to show the visible, remaining lines connected to this axis. Multiple axes could be brushed at the same time to set multiple filtering criteria from different variables. The order of an axis can be rearranged by dragging the label above each axis.

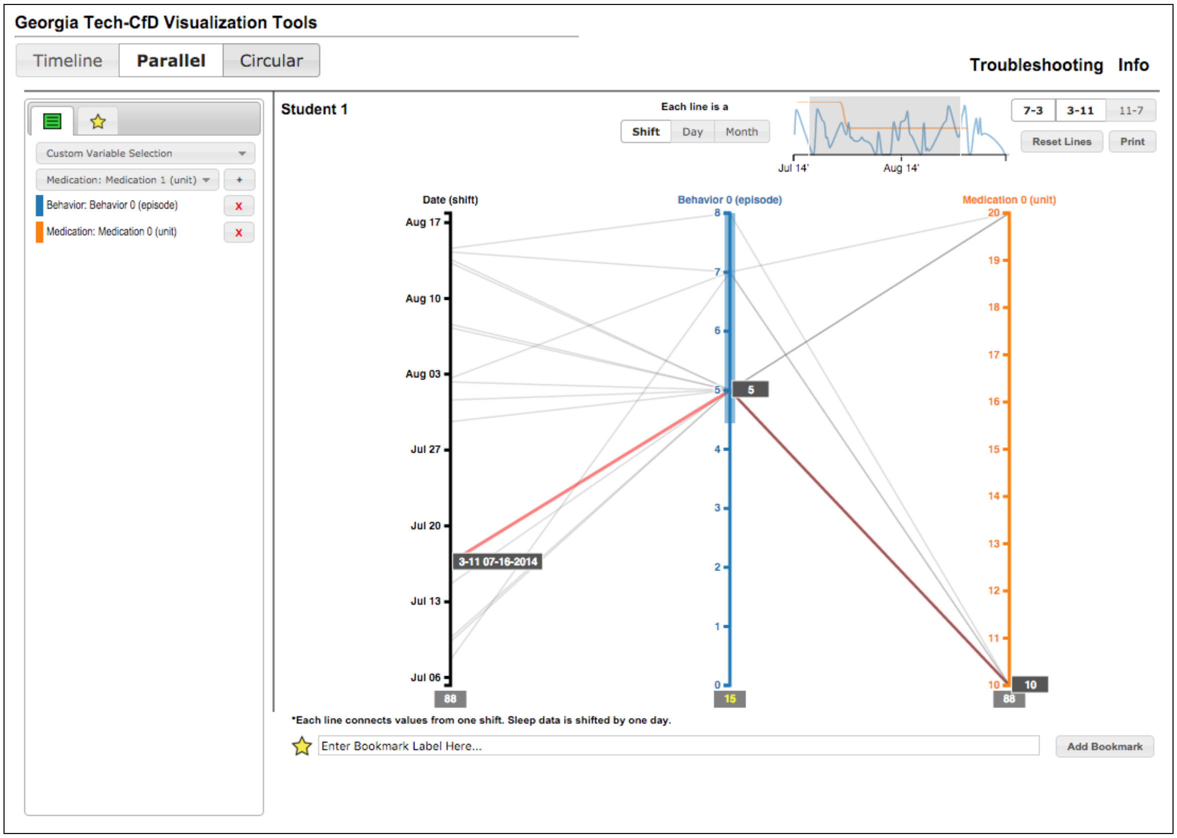


Figure 75: Screenshot of the Parallel view

The Parallel view was deployed in September 2015. A set of workshops were given to introduce it to the BAs. It provides a more effective way than the Timeline view for the BAs to find relations between variables. However, a limitation of parallel coordinates is that it only shows the relations between variables (axes) that are listed next to each other effectively. As a result, if more than three variables, including time, is shown, only a subset of pairwise relations can be easily observed. To address this issue, we designed the Circular view.

A.2.4 Circular view

The Circular view shows similar relational information to the Parallel view but uses a Circular layout where all relations between all the variables could be examined at the same time.

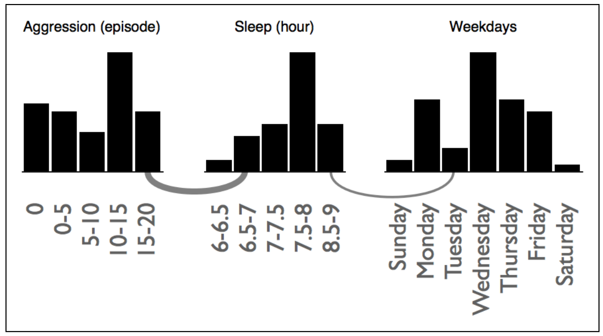


Figure 76: Basic visual representation of the Circular view

See Figure 76 for an illustration of the basic visual representation of the Circular view. Each bar chart is a frequency distribution of a variable over a set of value ranges (bins). A line is drawn between two bars if any of their data points occurred in the same time window (e.g., day). The width of the line shows the amount of the co-occurring data points. For example, if there were many days when a student slept less and had more aggressive behaviors, there will be a thick line (arc) connecting the two corresponding bars as shown between the Aggregation and Sleep charts in Figure 76. Similarly, if days with more sleep occurred on Tuesday, there will also be a line (arc) connecting the two bars as shown by the (relatively) thin line between the Sleep and Weekdays charts in Figure 76. Alternatively, the width of the line can be mapped to other statistics such as the percentage of co-occurrence. If we position the bar charts around a half circle and draw the connections, we have the Circular view, as shown in Figure 77. The Circular view was deployed alongside the Parallel view in September 2015.

In this view, interactive exploration is important to find valuable information. Using the interactive sliders, users can make various configurations to the visualization such as filtering the lines by width, removing line connections with the temporal variables in the bottom, or changing the temporal variables on the horizontal axis from the weekdays and shifts to months. Moving the mouse cursor over the labels of the bars or lines would highlight the co-occurrence amounts in different ways. One

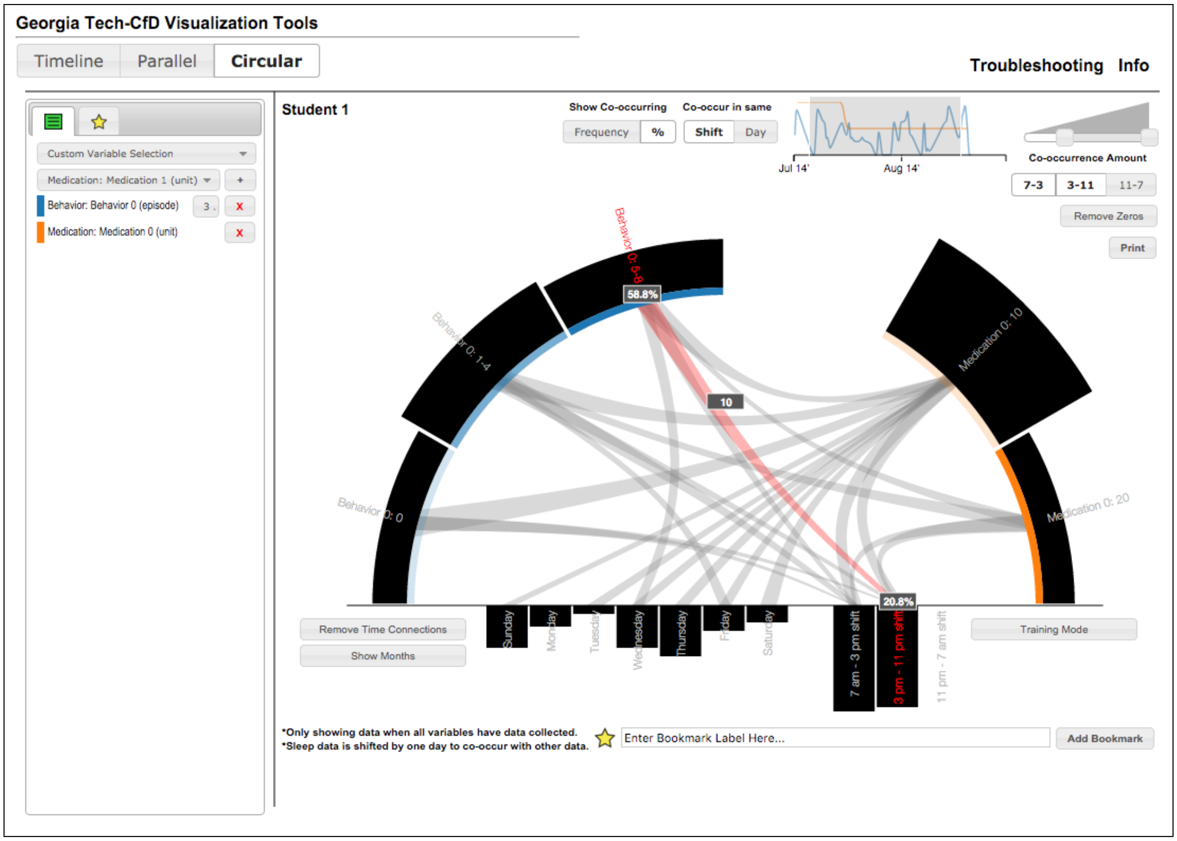


Figure 77: Screenshot of the Circular view

example of such highlights are shown in Figure 77. Because of the complexity of the visualization, a "training mode" is available to supply a descriptive explanation at the bottom of the view upon certain interactions.

A.3 Surveys and Interaction Logging

To collect usage information, I instrumented the application with surveys and interaction logging. These functions were designed carefully to minimize the burden on the CfD users as the application was planned to be deployed over a long period of time. The research study was approved by both the Georgia Tech and CfD IRBs.

A.3.1 Surveys

- Start survey (Figure 73)

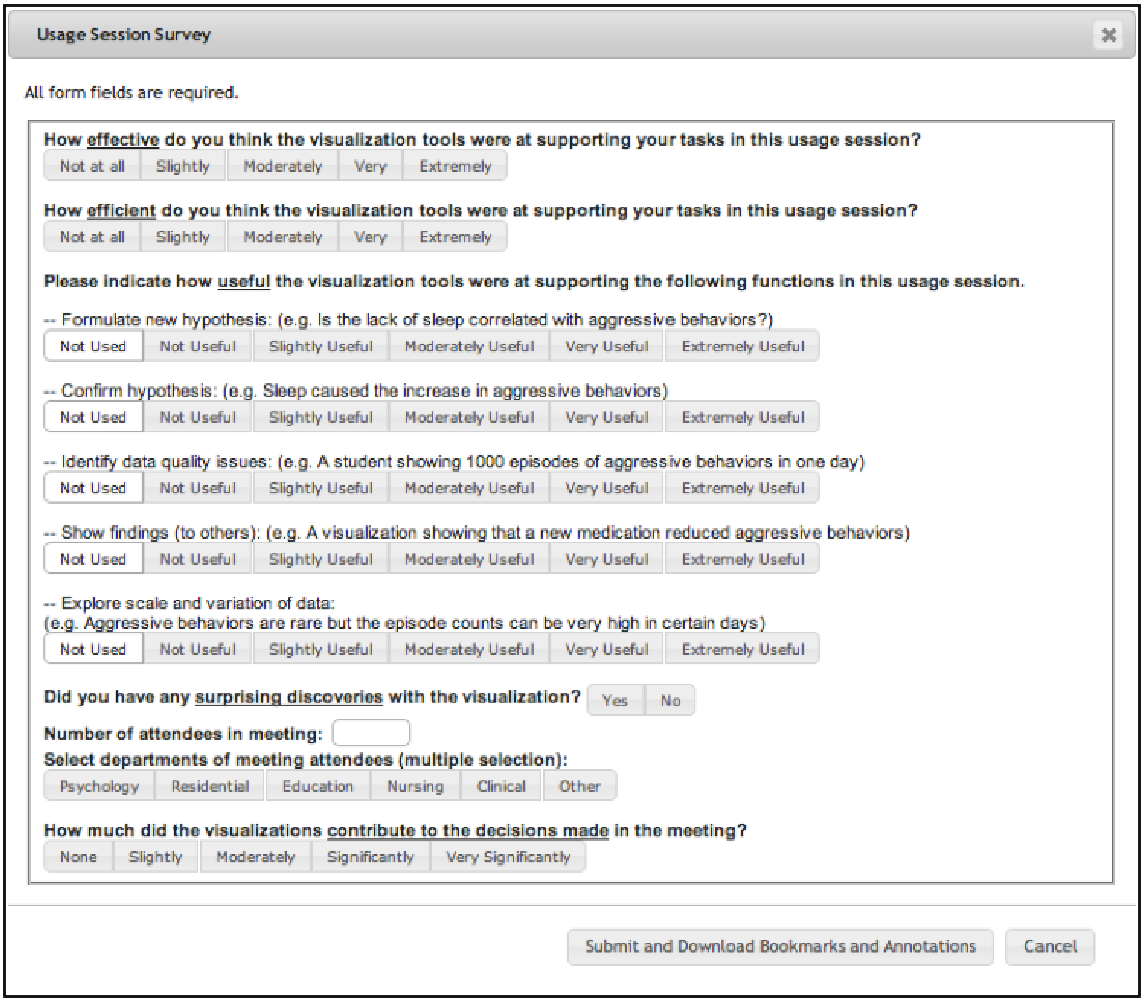
  The start survey collects information about the user and the usage reason (e.g., team meeting) for each session.

- End survey (Figure 78)

  The end survey is connected with exporting bookmarks and annotations. When users choose to export them, this survey will be presented. It prompts users to specify their subjective opinions about the application (e.g., perceived effectiveness) and whether they had any surprising discovery. This survey was connected to this feature as it is likely the last step in the visual analysis process. Depending on the usage reason, the survey could have different questions (Figure 78 shows questions for team meeting usages). Note that because not all usage sessions would lead to bookmarks and annotations that need to be saved for later reference, this survey is not always administered.

A.3.2 Interaction Logging

Interactions in the visualization application are logged in the way defined in Section 4.2. Chapter 4 figures have examples of logged interaction events from the CfD Visualization.



Usage Session Survey

All form fields are required.

How effective do you think the visualization tools were at supporting your tasks in this usage session?

Not at all | Slightly | Moderately | Very | Extremely

How efficient do you think the visualization tools were at supporting your tasks in this usage session?

Not at all | Slightly | Moderately | Very | Extremely

Please indicate how useful the visualization tools were at supporting the following functions in this usage session.

-- Formulate new hypothesis: (e.g. Is the lack of sleep correlated with aggressive behaviors?)

Not Used | Not Useful | Slightly Useful | Moderately Useful | Very Useful | Extremely Useful

-- Confirm hypothesis: (e.g. Sleep caused the increase in aggressive behaviors)

Not Used | Not Useful | Slightly Useful | Moderately Useful | Very Useful | Extremely Useful

-- Identify data quality issues: (e.g. A student showing 1000 episodes of aggressive behaviors in one day)

Not Used | Not Useful | Slightly Useful | Moderately Useful | Very Useful | Extremely Useful

-- Show findings (to others): (e.g. A visualization showing that a new medication reduced aggressive behaviors)

Not Used | Not Useful | Slightly Useful | Moderately Useful | Very Useful | Extremely Useful

-- Explore scale and variation of data:
(e.g. Aggressive behaviors are rare but the episode counts can be very high in certain days)

Not Used | Not Useful | Slightly Useful | Moderately Useful | Very Useful | Extremely Useful

Did you have any surprising discoveries with the visualization? Yes | No

Number of attendees in meeting:

Select departments of meeting attendees (multiple selection):

Psychology | Residential | Education | Nursing | Clinical | Other

How much did the visualizations contribute to the decisions made in the meeting?

None | Slightly | Moderately | Significantly | Very Significantly

Submit and Download Bookmarks and Annotations | Cancel

Figure 78: End survey (use in team meeting)

REFERENCES

[1] Alspaugh, S., Chen, B., Lin, J., Ganapathi, A., Hearst, M., and Katz, R., "Analyzing Log Analysis: An Empirical Study of User Log Mining," in *28th Large Installation System Administration Conference (LISA14)*, (Seattle, WA), pp. 62–77, USENIX Association, Nov. 2014.

[2] Andrews, C., Endert, A., and North, C., "Space to Think: Large High-resolution Displays for Sensemaking," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, (New York, NY, USA), pp. 55–64, ACM, 2010.

[3] Benevenuto, F., Rodrigues, T., Cha, M., and Almeida, V., "Characterizing user navigation and interactions in online social networks," *Information Sciences*, vol. 195, pp. 1–24, July 2012.

[4] Blascheck, T., John, M., Kurzhals, K., Koch, S., and Ertl, T., "VA2: A Visual Analytics Approach for Evaluating Visual Analytics Applications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 61–70, Jan. 2016.

[5] Brehmer, M. and Munzner, T., "A Multi-Level Typology of Abstract Visualization Tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 2376–2385, Dec. 2013.

[6] Brown, E., Ottley, A., Zhao, H., Lin, Q., Souvenir, R., Endert, A., and Chang, R., "Finding Waldo: Learning about Users from their Interactions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, pp. 1663–1672, Dec. 2014.

[7] Card, S. K., Mackinlay, J., and Shneiderman, B., eds., *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1 ed., Feb. 1999.

[8] Card, S. K., Pirolli, P., Van Der Wege, M., Morrison, J. B., Reeder, R. W., Schraedley, P. K., and Boshart, J., "Information Scent As a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, (New York, NY, USA), pp. 498–505, ACM, 2001.

[9] Catledge, L. D. and Pitkow, J. E., "Characterizing browsing strategies in the World-Wide web," *Computer Networks and ISDN Systems*, vol. 27, pp. 1065–1073, Apr. 1995.

[10] Chorianopoulos, K., "Collective intelligence within web video," *Human-centric Computing and Information Sciences*, vol. 3, pp. 1–16, June 2013.

[11] Coull, S., Branch, J., Szymanski, B., and Breimer, E., "Intrusion detection: a bioinformatics approach," in *Proceedings of the 19th Annual Computer Security Applications Conference*, pp. 24–33, Dec. 2003.

[12] Dou, W., Jeong, D. H., Stukes, F., Ribarsky, W., Lipford, H., and Chang, R., "Recovering Reasoning Processes from User Interactions," *IEEE Computer Graphics and Applications*, vol. 29, pp. 52–61, May 2009.

[13] Gotz, D. and Zhou, M., "Characterizing user's visual analytic activity for insight provenance," in *IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST '08*, pp. 123–130, Oct. 2008.

[14] Gotz, D. and Wen, Z., "Behavior-driven Visualization Recommendation," in *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, (New York, NY, USA), pp. 315–324, ACM, 2009.

[15] Gray, M., Badre, A., and Guzdial, M., "Visualizing usability log data," in *Proceedings IEEE Symposium on Information Visualization '96*, pp. 93–98, 130, Oct. 1996.

[16] Guo, H., Gomez, S., Ziemkiewicz, C., and Laidlaw, D., "A Case Study Using Visualization Interaction Logs and Insight Metrics to Understand How Analysts Arrive at Insights," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 51–60, Jan. 2016.

[17] Han, Y., Abowd, G. D., and Stasko, J., "Flexible Organization, Exploration, and Analysis of Visualization Application Interaction Events using Visual Analytics," in *IEEE VIS 2016 Workshop on The Event Event: Temporal & Sequential Event Analysis*, (Baltimore, MD, USA), 2016.

[18] Han, Y., Abowd, G. D., and Stasko, J., "Requirements for Visual Interaction Analysis Systems," in *IEEE VIS 2016 Workshop on Logging Interactive Visualizations & Visualizing Interaction Logs*, (Baltimore, MD, USA), 2016.

[19] Heer, J., Mackinlay, J., Stolte, C., and Agrawala, M., "Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1189–1196, Nov. 2008.

[20] Hilbert, D. M. and Redmiles, D. F., "Extracting Usability Information from User Interface Events," *ACM Computing Surveys*, vol. 32, pp. 384–421, Dec. 2000.

[21] Hollink, L., Mika, P., and Blanco, R., "Web Usage Mining with Semantic Analysis," in *Proceedings of the 22nd International Conference on World Wide*

Web, WWW '13, (Republic and Canton of Geneva, Switzerland), pp. 561–570, International World Wide Web Conferences Steering Committee, 2013.

[22] Iglesias, J., Angelov, P., Ledezma, A., and Sanchis, A., "Creating Evolving User Behavior Profiles Automatically," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, pp. 854–867, May 2012.

[23] Inselberg, A., "The plane with parallel coordinates," *The Visual Computer*, vol. 1, pp. 69–91, Aug. 1985.

[24] Jeong, D. H., Dou, W., Lipford, H., Stukes, F., Chang, R., and Ribarsky, W., "Evaluating the relationship between user interaction and financial visual analysis," in *IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST '08*, pp. 83–90, Oct. 2008.

[25] Kang, Y.-a., Gorg, C., and Stasko, J., "Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study," in *IEEE Symposium on Visual Analytics Science and Technology, 2009. VAST 2009*, pp. 139–146, Oct. 2009.

[26] Kim, J., Guo, P. J., Seaton, D. T., Mitros, P., Gajos, K. Z., and Miller, R. C., "Understanding In-video Dropouts and Interaction Peaks Inonline Lecture Videos," in *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, (New York, NY, USA), pp. 31–40, ACM, 2014.

[27] Koch, K., McLean, J., Segev, R., Freed, M. A., Berry, M. J., Balasubramanian, V., and Sterling, P., "How Much the Eye Tells the Brain," *Current Biology*, vol. 16, pp. 1428–1434, July 2006.

[28] Lam, H., Russell, D., Tang, D., and Munzner, T., "Session Viewer: Visual Exploratory Analysis of Web Session Logs," in *IEEE Symposium on Visual Analytics Science and Technology, 2007. VAST 2007*, pp. 147–154, Oct. 2007.

[29] Nguyen, P., Xu, K., Wheat, A., Wong, B., Attfield, S., and Fields, B., "SensePath: Understanding the Sensemaking Process Through Analytic Provenance," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 41–50, Jan. 2016.

[30] Nicholas, D., Huntington, P., and Watkinson, A., "Scholarly journal usage: the results of deep log analysis," *Journal of Documentation*, vol. 61, pp. 248–280, Apr. 2005.

[31] Pachidi, S., Spruit, M., and van de Weerd, I., "Understanding users behavior with software operation data mining," *Computers in Human Behavior*, vol. 30, pp. 583–594, Jan. 2014.

[32] Pepyne, D., Hu, J., and Gong, W., "User profiling for computer security," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2, pp. 982–987 vol.2, June 2004.

[33] Pitkow, J. E. and Bharat, K. A., “WEBVIZ: A Tool for World Wide Web Access Log Analysis,” technical Report, GVU Center, Georgia Institute of Technology, 1994.

[34] Pohl, M., Wiltner, S., Miksch, S., Aigner, W., and Rind, A., “Analysing Interactivity in Information Visualisation,” *KI - Knstliche Intelligenz*, vol. 26, pp. 151–159, May 2012.

[35] Reda, K., Johnson, A. E., Leigh, J., and Papka, M. E., “Evaluating User Behavior and Strategy During Visual Exploration,” in *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, BELIV ’14, (New York, NY, USA), pp. 41–45, ACM, 2014.

[36] Rind, A., Aigner, W., Wagner, M., Miksch, S., and Lammarsch, T., “Task Cube: A three-dimensional conceptual space of user tasks in visualization design and evaluation,” *Information Visualization*, vol. 15, pp. 288–300, Oct. 2016.

[37] Schneider, F., Feldmann, A., Krishnamurthy, B., and Willinger, W., “Understanding Online Social Network Usage from a Network Perspective,” in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC ’09, (New York, NY, USA), pp. 35–48, ACM, 2009.

[38] Schonlau, M., DuMouchel, W., Ju, W.-H., Karr, A. F., Theusan, M., and Vardi, Y., “Computer Intrusion: Detecting Masquerades,” *Statistical Science*, vol. 16, pp. 58–74, Feb. 2001.

[39] Shen, Z., Wei, J., Sundaresan, N., and Ma, K.-L., “Visual analysis of massive web session data,” in *2012 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 65–72, Oct. 2012.

[40] Shi, C., Fu, S., Chen, Q., and Qu, H., “VisMOOC: Visualizing video clickstream data from Massive Open Online Courses,” in *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 159–166, Apr. 2015.

[41] Shneiderman, B., “The eyes have it: a task by data type taxonomy for information visualizations,” in *, IEEE Symposium on Visual Languages, 1996. Proceedings*, pp. 336–343, Sept. 1996.

[42] Sinha, T., Jermann, P., Li, N., and Dillenbourg, P., “Your click decides your fate: Inferring Information Processing and Attrition Behavior from MOOC Video Clickstream Interactions,” in *2014 Empirical Methods in Natural Language Processing Workshop on Modeling Large Scale Social Interaction in Massively Open Online Courses*, 2014.

[43] Spiliopoulou, M. and Pohle, C., “Data Mining for Measuring and Improving the Success of Web Sites,” *Data Mining and Knowledge Discovery*, vol. 5, pp. 85–114, Jan. 2001.

[44] Stasko, J., Grg, C., and Liu, Z., “Jigsaw: Supporting Investigative Analysis Through Interactive Visualization,” *Information Visualization*, vol. 7, pp. 118–132, Apr. 2008.

[45] Treisman, A., “Preattentive processing in vision,” *Computer Vision, Graphics, and Image Processing*, vol. 31, pp. 156–177, Aug. 1985.

[46] van Ham, F. and Perer, A., “”Search, Show Context, Expand on Demand”: Supporting Large Graph Exploration with Degree-of-Interest,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 953–960, Nov. 2009.

[47] Wei, J., Shen, Z., Sundaresan, N., and Ma, K.-L., “Visual cluster exploration of web clickstream data,” in *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 3–12, Oct. 2012.

[48] Weichbroth, P., Owoc, M., and Pleszkun, M., “Web user navigation patterns discovery from WWW server log files,” in *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 1171–1176, Sept. 2012.

[49] Wongsuphasawat, K. and Lin, J., “Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at Twitter,” in *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 113–122, Oct. 2014.

[50] Yi, J. S., Kang, Y. a., Stasko, J., and Jacko, J., “Toward a Deeper Understanding of the Role of Interaction in Information Visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1224–1231, Nov. 2007.

[51] Zaïane, O., Xin, M., and Han, J., “Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs,” in *IEEE International Forum on Research and Technology Advances in Digital Libraries, 1998. ADL 98. Proceedings*, pp. 19–29, Apr. 1998.

[52] Zhao, J., Liu, Z., Dontcheva, M., Hertzmann, A., and Wilson, A., “MatrixWave: Visual Comparison of Event Sequence Data,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, (New York, NY, USA), pp. 259–268, ACM, 2015.

REFERENCES

[1] Alspaugh, S., Chen, B., Lin, J., Ganapathi, A., Hearst, M., and Katz, R., "Analyzing Log Analysis: An Empirical Study of User Log Mining," in *28th Large Installation System Administration Conference (LISA14)*, (Seattle, WA), pp. 62–77, USENIX Association, Nov. 2014.

[2] Andrews, C., Endert, A., and North, C., "Space to Think: Large High-resolution Displays for Sensemaking," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, (New York, NY, USA), pp. 55–64, ACM, 2010.

[3] Benevenuto, F., Rodrigues, T., Cha, M., and Almeida, V., "Characterizing user navigation and interactions in online social networks," *Information Sciences*, vol. 195, pp. 1–24, July 2012.

[4] Blascheck, T., John, M., Kurzhals, K., Koch, S., and Ertl, T., "VA2: A Visual Analytics Approach for Evaluating Visual Analytics Applications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 61–70, Jan. 2016.

[5] Brehmer, M. and Munzner, T., "A Multi-Level Typology of Abstract Visualization Tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, pp. 2376–2385, Dec. 2013.

[6] Brown, E., Ottley, A., Zhao, H., Lin, Q., Souvenir, R., Endert, A., and Chang, R., "Finding Waldo: Learning about Users from their Interactions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, pp. 1663–1672, Dec. 2014.

[7] Card, S. K., Mackinlay, J., and Shneiderman, B., eds., *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1 ed., Feb. 1999.

[8] Card, S. K., Pirolli, P., Van Der Wege, M., Morrison, J. B., Reeder, R. W., Schraedley, P. K., and Boshart, J., "Information Scent As a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, (New York, NY, USA), pp. 498–505, ACM, 2001.

[9] Catledge, L. D. and Pitkow, J. E., "Characterizing browsing strategies in the World-Wide web," *Computer Networks and ISDN Systems*, vol. 27, pp. 1065–1073, Apr. 1995.

[10] Chorianopoulos, K., "Collective intelligence within web video," *Human-centric Computing and Information Sciences*, vol. 3, pp. 1–16, June 2013.

[11] Coull, S., Branch, J., Szymanski, B., and Breimer, E., "Intrusion detection: a bioinformatics approach," in *Proceedings of the 19th Annual Computer Security Applications Conference*, pp. 24–33, Dec. 2003.

[12] Dou, W., Jeong, D. H., Stukes, F., Ribarsky, W., Lipford, H., and Chang, R., "Recovering Reasoning Processes from User Interactions," *IEEE Computer Graphics and Applications*, vol. 29, pp. 52–61, May 2009.

[13] Gotz, D. and Zhou, M., "Characterizing user's visual analytic activity for insight provenance," in *IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST '08*, pp. 123–130, Oct. 2008.

[14] Gotz, D. and Wen, Z., "Behavior-driven Visualization Recommendation," in *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, (New York, NY, USA), pp. 315–324, ACM, 2009.

[15] Gray, M., Badre, A., and Guzdial, M., "Visualizing usability log data," in *Proceedings IEEE Symposium on Information Visualization '96*, pp. 93–98, 130, Oct. 1996.

[16] Guo, H., Gomez, S., Ziemkiewicz, C., and Laidlaw, D., "A Case Study Using Visualization Interaction Logs and Insight Metrics to Understand How Analysts Arrive at Insights," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 51–60, Jan. 2016.

[17] Han, Y., Abowd, G. D., and Stasko, J., "Flexible Organization, Exploration, and Analysis of Visualization Application Interaction Events using Visual Analytics," in *IEEE VIS 2016 Workshop on The Event Event: Temporal & Sequential Event Analysis*, (Baltimore, MD, USA), 2016.

[18] Han, Y., Abowd, G. D., and Stasko, J., "Requirements for Visual Interaction Analysis Systems," in *IEEE VIS 2016 Workshop on Logging Interactive Visualizations & Visualizing Interaction Logs*, (Baltimore, MD, USA), 2016.

[19] Heer, J., Mackinlay, J., Stolte, C., and Agrawala, M., "Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1189–1196, Nov. 2008.

[20] Hilbert, D. M. and Redmiles, D. F., "Extracting Usability Information from User Interface Events," *ACM Computing Surveys*, vol. 32, pp. 384–421, Dec. 2000.

[21] Hollink, L., Mika, P., and Blanco, R., "Web Usage Mining with Semantic Analysis," in *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, (Republic and Canton of Geneva, Switzerland), pp. 561–570, International World Wide Web Conferences Steering Committee, 2013.

[22] Iglesias, J., Angelov, P., Ledezma, A., and Sanchis, A., "Creating Evolving User Behavior Profiles Automatically," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, pp. 854–867, May 2012.

[23] Inselberg, A., "The plane with parallel coordinates," *The Visual Computer*, vol. 1, pp. 69–91, Aug. 1985.

[24] Jeong, D. H., Dou, W., Lipford, H., Stukes, F., Chang, R., and Ribarsky, W., "Evaluating the relationship between user interaction and financial visual analysis," in *IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST '08*, pp. 83–90, Oct. 2008.

[25] Kang, Y.-a., Gorg, C., and Stasko, J., "Evaluating visual analytics systems for investigative analysis: Deriving design principles from a case study," in *IEEE Symposium on Visual Analytics Science and Technology, 2009. VAST 2009*, pp. 139–146, Oct. 2009.

[26] Kim, J., Guo, P. J., Seaton, D. T., Mitros, P., Gajos, K. Z., and Miller, R. C., "Understanding In-video Dropouts and Interaction Peaks Inonline Lecture Videos," in *Proceedings of the First ACM Conference on Learning @ Scale Conference*, L@S '14, (New York, NY, USA), pp. 31–40, ACM, 2014.

[27] Koch, K., McLean, J., Segev, R., Freed, M. A., Berry, M. J., Balasubramanian, V., and Sterling, P., "How Much the Eye Tells the Brain," *Current Biology*, vol. 16, pp. 1428–1434, July 2006.

[28] Lam, H., Russell, D., Tang, D., and Munzner, T., "Session Viewer: Visual Exploratory Analysis of Web Session Logs," in *IEEE Symposium on Visual Analytics Science and Technology, 2007. VAST 2007*, pp. 147–154, Oct. 2007.

[29] Nguyen, P., Xu, K., Wheat, A., Wong, B., Attfield, S., and Fields, B., "SensePath: Understanding the Sensemaking Process Through Analytic Provenance," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 41–50, Jan. 2016.

[30] Nicholas, D., Huntington, P., and Watkinson, A., "Scholarly journal usage: the results of deep log analysis," *Journal of Documentation*, vol. 61, pp. 248–280, Apr. 2005.

[31] Pachidi, S., Spruit, M., and van de Weerd, I., "Understanding users behavior with software operation data mining," *Computers in Human Behavior*, vol. 30, pp. 583–594, Jan. 2014.

[32] Pepyne, D., Hu, J., and Gong, W., "User profiling for computer security," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2, pp. 982–987 vol.2, June 2004.

[33] Pitkow, J. E. and Bharat, K. A., "WEBVIZ: A Tool for World Wide Web Access Log Analysis," technical Report, GVU Center, Georgia Institute of Technology, 1994.

[34] Pohl, M., Wiltner, S., Miksch, S., Aigner, W., and Rind, A., "Analysing Interactivity in Information Visualisation," *KI - Knstliche Intelligenz*, vol. 26, pp. 151–159, May 2012.

[35] Reda, K., Johnson, A. E., Leigh, J., and Papka, M. E., "Evaluating User Behavior and Strategy During Visual Exploration," in *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, BELIV '14, (New York, NY, USA), pp. 41–45, ACM, 2014.

[36] Rind, A., Aigner, W., Wagner, M., Miksch, S., and Lammarsch, T., "Task Cube: A three-dimensional conceptual space of user tasks in visualization

design and evaluation," *Information Visualization*, vol. 15, pp. 288–300, Oct. 2016.

[37] Schneider, F., Feldmann, A., Krishnamurthy, B., and Willinger, W., "Understanding Online Social Network Usage from a Network Perspective," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, (New York, NY, USA), pp. 35–48, ACM, 2009.

[38] Schonlau, M., DuMouchel, W., Ju, W.-H., Karr, A. F., Theusan, M., and Vardi, Y., "Computer Intrusion: Detecting Masquerades," *Statistical Science*, vol. 16, pp. 58–74, Feb. 2001.

[39] Shen, Z., Wei, J., Sundaresan, N., and Ma, K.-L., "Visual analysis of massive web session data," in *2012 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 65–72, Oct. 2012.

[40] Shi, C., Fu, S., Chen, Q., and Qu, H., "VisMOOC: Visualizing video clickstream data from Massive Open Online Courses," in *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 159–166, Apr. 2015.

[41] Shneiderman, B., "The eyes have it: a task by data type taxonomy for information visualizations," in *, IEEE Symposium on Visual Languages, 1996. Proceedings*, pp. 336–343, Sept. 1996.

[42] Sinha, T., Jermann, P., Li, N., and Dillenbourg, P., "Your click decides your fate: Inferring Information Processing and Attrition Behavior from MOOC Video Clickstream Interactions," in *2014 Empirical Methods in Natural Language Processing Workshop on Modeling Large Scale Social Interaction in Massively Open Online Courses*, 2014.

[43] Spiliopoulou, M. and Pohle, C., "Data Mining for Measuring and Improving the Success of Web Sites," *Data Mining and Knowledge Discovery*, vol. 5, pp. 85–114, Jan. 2001.

[44] Stasko, J., Grg, C., and Liu, Z., "Jigsaw: Supporting Investigative Analysis Through Interactive Visualization," *Information Visualization*, vol. 7, pp. 118–132, Apr. 2008.

[45] Treisman, A., "Preattentive processing in vision," *Computer Vision, Graphics, and Image Processing*, vol. 31, pp. 156–177, Aug. 1985.

[46] van Ham, F. and Perer, A., ""Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 953–960, Nov. 2009.

[47] Wei, J., Shen, Z., Sundaresan, N., and Ma, K.-L., "Visual cluster exploration of web clickstream data," in *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 3–12, Oct. 2012.

[48] Weichbroth, P., Owoc, M., and Pleszkun, M., "Web user navigation patterns discovery from WWW server log files," in *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 1171–1176, Sept. 2012.

[49] Wongsuphasawat, K. and Lin, J., "Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at Twitter," in *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 113–122, Oct. 2014.

[50] Yi, J. S., Kang, Y. a., Stasko, J., and Jacko, J., "Toward a Deeper Understanding of the Role of Interaction in Information Visualization," *IEEE*

Transactions on Visualization and Computer Graphics, vol. 13, pp. 1224–1231, Nov. 2007.

[51] Zaïane, O., Xin, M., and Han, J., "Discovering Web access patterns and trends by applying OLAP and data mining technology on Web logs," in *IEEE International Forum on Research and Technology Advances in Digital Libraries, 1998. ADL 98. Proceedings*, pp. 19–29, Apr. 1998.

[52] Zhao, J., Liu, Z., Dontcheva, M., Hertzmann, A., and Wilson, A., "MatrixWave: Visual Comparison of Event Sequence Data," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, (New York, NY, USA), pp. 259–268, ACM, 2015.